NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

S1587

A FEASIBILITY STUDY IN
PATH PLANNING USING
OPTIMIZATION TECHNIQUES

by

David W. Sanders

December 1987

Thesis Advisor David L. Smith

Approved for public release; distribution is unlimited

T239273

REPORT DOCUMENTATION PAGE

a. REPORT SECURITY CLASSIFICATION unclassified	1b RESTRICTIVE MARKINGS
a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
b DECLASSIFICATION / DOWNGRADING SCHEDULE	
PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 69
7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
NAME OF FUNDING / SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)
9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

TITLE (Include Security Classification)
FEASIBILITY STUDY IN PATH PLANNING APPLICATIONS USING OPTIMIZATION ECHNIQUES

PERSONAL AUTHOR(S)
Sanders, David W.

a. TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
aster's Thesis	FROM _____ TO _____	1987 December	91

SUPPLEMENTARY NOTATION

COSATI CODES		
FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Path Planning, Optimization, Optimum Control Theory, Autonomous Underwater Vehicle

9 ABSTRACT (Continue on reverse if necessary and identify by block number)

Path Planning is an intricate part of the navigation function of any vehicle traveling between two points in space. In an autonomous underwater vehicle, a trajectory may be planned between two points using the optimization techniques of ADS (Advanced Design Synthesis) coupled to a motion analysis routine, DSL (Dynamic Simulation Language). The problem is posed as a two-point-boundary-value problem with initial states and desired final states known, as well as a final time specified. The objective function is minimized in the form of a quadratic regulator for the purpose of conserving the vehicles energy supply. An obstacle in the dive plane (X-Z plane) is introduced and successfully avoided using the constrained optimization

0 DISTRIBUTION / AVAILABILITY OF ABSTRACT 1 UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION unclassified
2 NAME OF RESPONSIBLE INDIVIDUAL avid L. Smith	22b TELEPHONE (Include Area Code) 408-646-3383 / 22c OFFICE SYMBOL 69Sm

19. ABSTRACT (continued)

technique. The use of optimization is proven as a feasible method for successfully planning the trajectories of underwater vehicles.

Approved for public release; distribution is unlimited

A Feasibility Study in Path Planning Applications Using Optimization Techniques

by

David W. Sanders
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1987

587

ABSTRACT

Path Planning is an intricate part of the navigation function of any vehicle traveling between two points in space. In an autonomous underwater vehicle, a trajectory may be planned between two points using the optimization techniques of ADS (Advanced Design Synthesis) coupled to a motion analysis routine, DSL (Dynamic Simulation Language). The problem is posed as a two-point-boundary-value problem with initial states and desired final states known, as well as a final time specified. The objective function is minimized in the form of a quadratic regulator for the purpose of conserving the vehicles energy supply. An obstacle in the dive plane (X-Z plane) is introduced and successfully avoided using the constrained optimization technique. The use of optimization is proven as a feasible method for successfully planning the trajectories of underwater vehicles.

TABLE OF CONTENTS

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

ACKNOWLEDGEMENTS

The author would like to extend his heartfelt appreciation and thanks to Associate Professor David L. Smith for his continuing encouragement and enthusiasm expressed during the conduct of this research.

A special and well deserved note of appreciation is extended to my wife, Barbara, and our children, Jennifer and Julie, for their enduring support throughout the many months of thesis research.

I. INTRODUCTION

Path planning is the function provided by an intelligent system which determines a safe, collision free trajectory of travel between two points, a start point and a target point, for a specific time lapse. This has been performed successfully for a number of land vehicles by many different techniques [Refs.1-3]. Several classes of techniques available today include graphical search methods [Refs.4-8], artificial potential field methods [Refs. 9-13] and optimal control theory [Refs. 14,15]. The approach taken in the present work falls in the optimal control theory class. Here, a single-valued penalty function was used to evaluate a path between the two positions. The "best" path was then found by minimizing the penalty. The mathematics of this approach was fairly intense, but the advantage is that optimization in space and time were accomplished simultaneously.

Path planning is an open loop control problem that optimizes the control vector U to produce a best state trajectory, X, based on the penalty function, which is also referred to as the objective function.

This baseline study of optimal control theory as applied to path planning was directed toward being utilized in an Autonomous Underwater Vehicle (AUV) testbed at the Naval

Postgraduate School. At the time of this writing an AUV was being designed to operate in the ocean environment untethered from any command structure or man-in-the-loop system. To operate on its own the AUV must have, in addition to many other self sustaining systems, a system or means to plan its track from the present position to a target position some distance away and at some specified time in the future. Figure 1.1 shows the states that are of major concern when dealing with motion in the vertical plane. The x-state is the distance the vehicle travels in the horizontal direction. The z-state is the distance

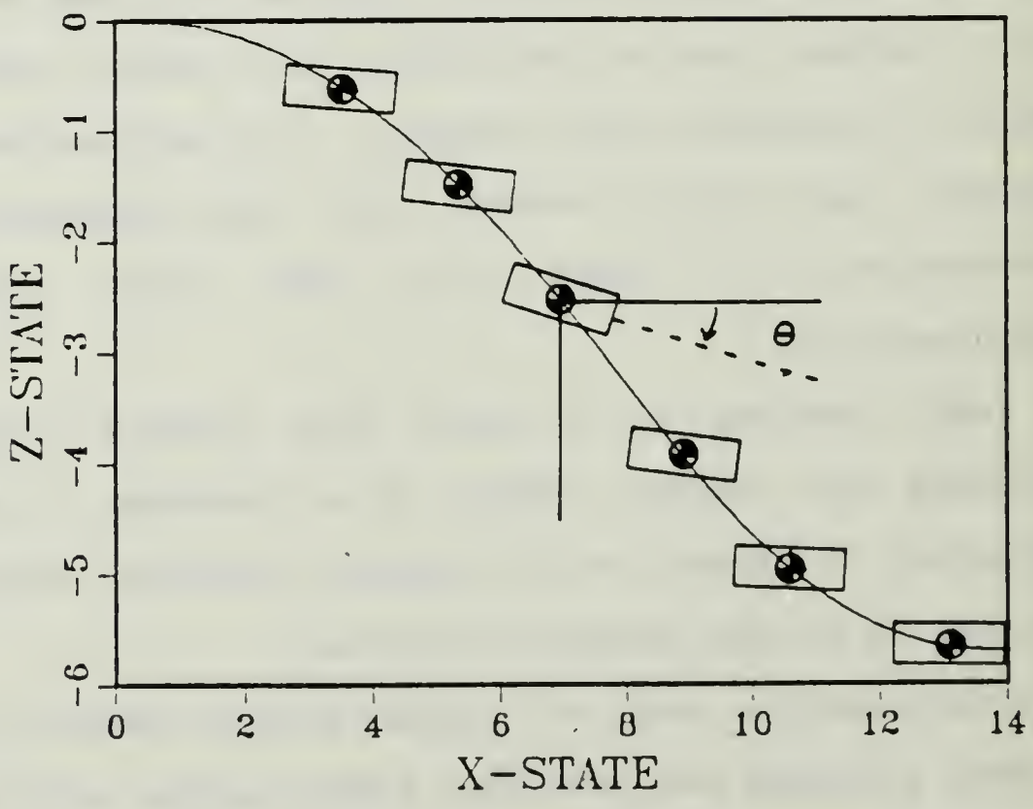


Figure 1.1 Dive Plane State Relationship

the vehicle travels vertically. The theta-state is the pitch angle of the vehicle, in radians. As mentioned earlier, the objective was to have this path be safe, collision free and energy efficient. Energy utilization was of key importance since the power source was assumed to be limited to a finite source onboard. The path planner was assumed to operate in a large area transit mode where the speed would be moderate and the obstacle environment sparse.

The Automatic Design Synthesis (ADS) FORTRAN program [Ref. 16] was utilized to accomplish the optimization when coupled to the Dynamic Simulation Language (DSL) [Ref.17]. The coupled package was referred to by the acronym ADSL. The IBM 3033 mainframe computer system provided the environment to run the programs. ADS provided an options package that allowed for the utilization of all known methods available to do numerical optimization. DSL provided selection of integration methods necessary to integrate the equations of motion (state equation) and determine the "best" path. The amount of time necessary to compute this path was also a factor worthy of consideration as it was desired to have the path planner perform at, or near, real time. It was assumed that these algorithms would later be hard-wired into a system for use in an underwater vehicle.

The approach taken was to initially utilize ADSL on an unconstrained nonlinear SISO (single input single output)

open loop minimization problem. The problem was chosen for its similarity to the AUV nonlinear dynamics structure. This problem allowed for initial studies on effectiveness of integrators, time interval step sizes, strategy and optimizers. This procedure was then applied to a simplified linear model of dive plane motion with a corresponding multiple input multiple output (MIMO) model. Comparison of results with the full-scale nonlinear model developed by Lt. Boncal followed [Ref.18]. The "best" path was determined in all cases.

II. SISO MINIMIZATION PROBLEM

A. THE PROBLEM

The SISO minimization problem was based on the continuous time problem of example 10.2-6 of Sage [Ref.19:p.313] (see Appendix A). This problem was chosen because of its similarity to the structure of the nonlinear dynamics of an underwater vehicle which will be discussed in the next part of this study. Sage arrived at the solution by using a continuous time analytical method and a gradient-in-function-space technique. Optimization was reached through an analytical equation that gave the results plotted in Figure 2.1 for the control variable, and Figure 2.2 for the state variable.

In the present work a discrete time solution was chosen that gave a set of single-valued control inputs evenly distributed over a specified number of delta-time steps. The optimum open loop control vector therefore appeared as a stepped function in time, which well mimics a computer-based controller output.

B. OBJECTIVE FUNCTION

The objective function was a one dimensional quadratic performance index of the form:

$$J = \int_0^T (X(t)^T Q X(t) + U(t)^T R U(t)) dt \qquad (2\text{-}1)$$

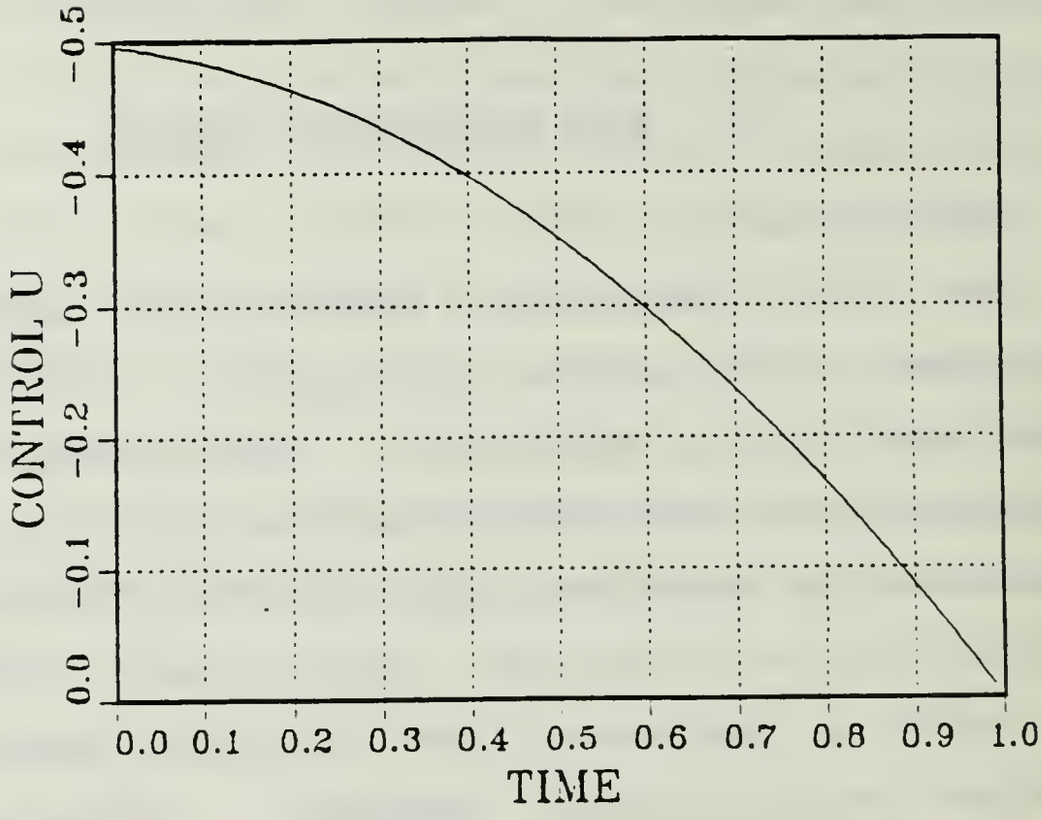


Figure 2.1 Control Optimization, after Sage [Ref. 19]

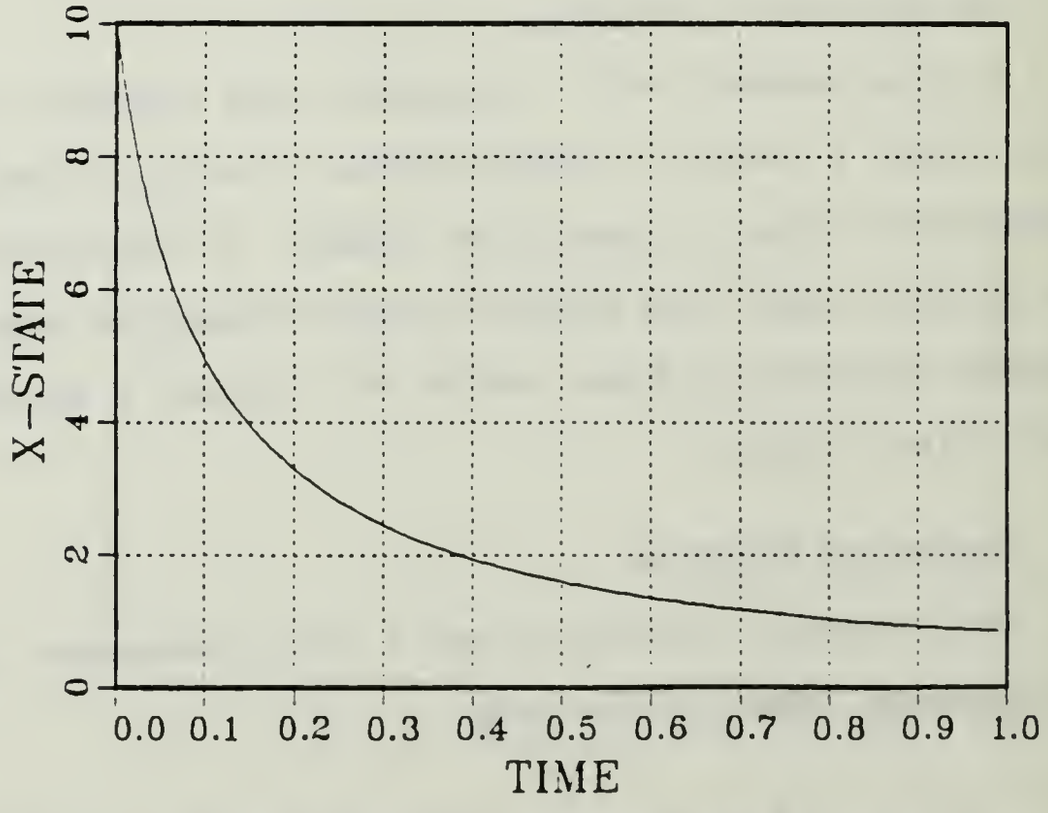


Figure 2.2 State Optimization, after Sage [Ref. 19]

where:

U = the control variable; and

X = the output state.

In actuality the computer saw the objective function as:

$$J = \sum_{1}^{N} (X_i^2 + U_i^2)\ \Delta T \qquad (2\text{-}2)$$

where:

N = the number of discrete time intervals, each of which corresponds to a single design variable (DV) in the control variable history.

Figure 2.3 shows the relationship between the design variables (DV), the time intervals, the discrete control variables (also DV) and the continuous control signal.

C. DESIGN VARIABLES

It was desired to chose a number of DV's in U that would yield a close approximation of the continuous time solution. Table 1 shows the mainframe computer run times and error data for cases where 100, 50, 20, and 10 time intervals were established between start time and finish time (0 to 1.0 sec). The rectangular integration method with .005 second step size was used for all cases.

Based on the extended virtual machine time (VM) on the NPS IBM mainframe computer for the 100 and 50 time interval

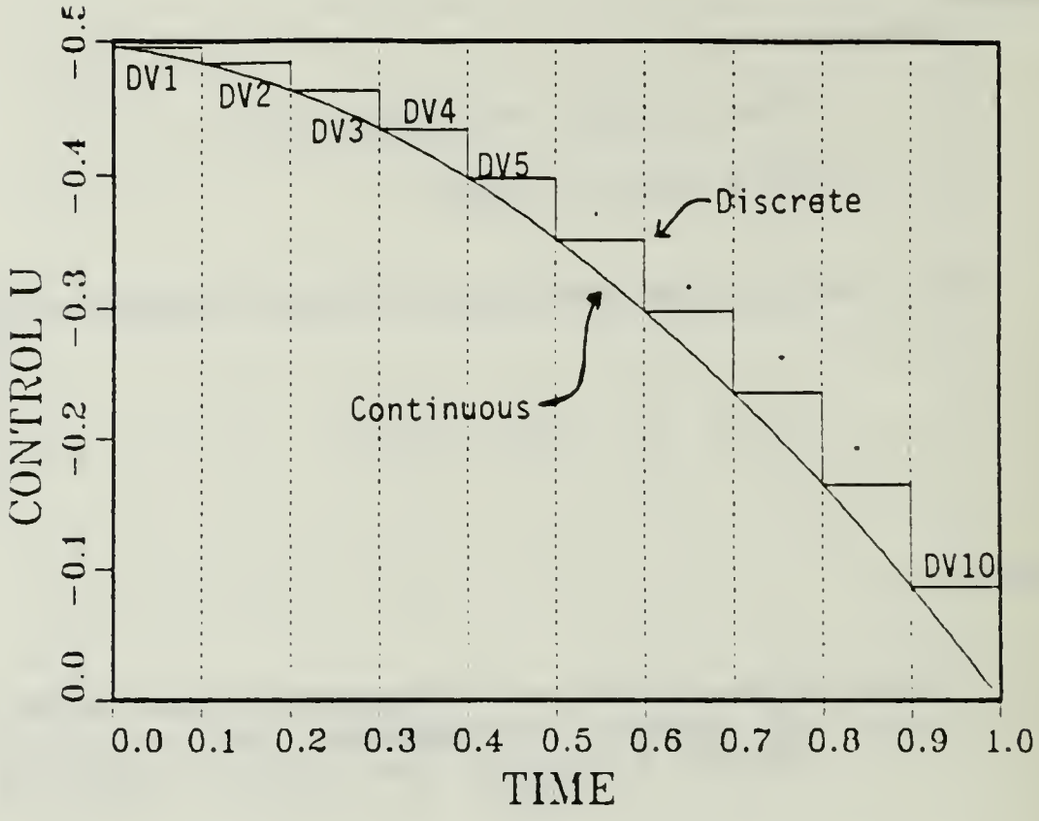


Figure 2.3 Discrete vs. Continuous Controls

TABLE 1

DESIGN VARIABLE COMPARISON

Number of Design Variable	Run Time	Objective	Objective Error	X-State Final Value	X-State Final Value Error
SAGE	-	4.4859	-	0.83327	-
100	8.38	4.4930	.16%	0.84347	1.2%
50	4.96	4.4934	.17%	0.84339	1.2%
20	3.30	4.4920	.14%	0.83680	.42%
10	2.46	4.4917	.13%	0.83385	.07%

problems and the fact that the results do not show any improvement in the solution, these two intervals were not considered as viable solutions. The 20 and 10 interval results were further analyzed to determine the number of design variables and also the recommended time step size to be used during the integration of both the state X and objective function J. First, however, the type of integration method needed to be determined.

D. INTEGRATION METHOD

DSL (Dynamic Simulation Language) has a variety of integrators to use for any type of problem. Table 2 gives the available methods in three groups.

TABLE 2

DSL INTEGRATION METHODS

Fixed Step Methods		Order
RECT	Rectangular	0
TRAPZ	Trapezoidal	1
SIMP	Simpson's	3
RKSFX	Runge-Kutta	4
Variable-step Methods		
RKS	Runge-Kutta	4
RK5	Runge-Kutta	5
Variable-step, variable-order Methods		
ADAMS	Adams-Moulton	1-12
STIFF	Gear, full Jacobian	1-5
BSTIFF	Gear, Banded Jacobian	1-5

The fixed step methods are easily controlled using the DELT parameter that tells DSL the delta time step size to use in the simulation. The rectangular method is the simplest, and the Runge-Kutta the more complex of the fixed step integration methods. Since the design variables were actually based on fixed time increments, the variable step methods were more difficult to utilize. However, with the additional DSL parameters of DELMAX and DELMIN these methods were also utilized to conduct the simulation comparisons. Comparisons using the 10 and 20 time intervals were made with the RECT, RKSFX, RK5, and ADAMS methods.

Results found in Table 3 give the run times and accuracies of both the objective function and the value of the X state at 1.0 second, the final time. The results show that the fixed step rectangular integration method was the overall most accurate method when considering both objective function and final X state after the 1.0 second interval. Also, as could be expected, the run times increased with complexity of the integration routines. To ensure that the variable step size methods covered each design variable time increment, the DELMAX and DELMIN values were set equal to the DELT parameter. The accuracies of the objective function were very good for the higher order integration methods; however, their accuracies for the final X state were worse than for the simple rectangular method.

TABLE 3

INTEGRATION METHOD RESULTS

	10 Design Variables			20 Design Variables		
Method	Run Time	Objective Error %	X-State Error %	Run Time	Objective Error %	X-State Error %
RECT	2.35	4.4917/.13	.83385/.07	2.16	4.4975/.26	.82678/.78
RKSFX	3.93	4.4863/.009	.84379/1.26	3.52	4.4866/.016	.84674/1.6
RK5	6.09	4.4863/.009	.84379/1.26	5.20	4.4866/.016	.84674/1.6
ADAMS	7.85	4.4863/.009	.84417/1.3	8.22	4.4881/.05	.85100/2.1
SAGE Results	-	4.4859	.83327	-	4.4859	.83327

When run time was considered as well, the "best for least" concept was used and the rectangular integration routine was selected as the best overall choice.

E. INTEGRATION STEP SIZE

The integration algorithms required a finite amount of time to run depending on step size. As the step size decreased, the time to complete the integration of the objective and state equations increased. Also, the accuracy of the integration result increased to a point then remained relatively constant. Table 4 provides tabulated data on integration step size, time requirements and accuracies for the two feasible time interval solutions.

TABLE 4

INTEGRATION RESULTS

Category	Time Intervals	20 DV's	Error %	10 DV's	Error %	Sage Result
Run	.001	11.38	-	7.61	-	-
	.005	3.30	-	2.46	-	
Time	.01	2.21	-	1.77	-	
(Sec)	.05	1.34	-	1.21	-	
	.001	4.4877	.04	4.4874	.03	4.4859
Objective	.005	4.4920	.14	4.4917	.13	
	.01	4.4975	.26	4.4972	.25	
Value	.05	4.5463	1.3	4.5463	1.3	
	.001	.84476	1.4	.84181	1.02	.83327
Final	.005	.83680	.42	.83385	.07	
X	.01	.82678	.78	.82383	1.13	
State	.05	.73755	11.5	.73648	11.6	

Results revealed that for 10 time intervals the best performance was found at the 0.005 second integration time step size. The run time was approximately equivalent to the 20 interval 0.01 second integration step size. The key to performance was the accuracy of the output state with respect to time. Figure 2.4 superimposes the Sage solution for the X state with the X state for the 20 interval 0.005 step size and the 10 interval 0.005 step size. They are all very close. However, the control prediction becomes the discriminating factor. Here, the 10 interval 0.005 step size shown in Figure 2.5 is the best choice for duplicating the control variable.

Therefore the overall "best for least" result was found to be the 10 time intervals, 0.005 second integration step size in conjunction with the rectangular integration algorithm.

F. OPTIMIZATION

The ADS (Advanced Design Synthesis) program provided the means to combine a wide variety of optimization concepts so that the best combination could be selected to solve the problem. The concepts were divided into three basic levels for solving the type of minimization problem described in Reference 20, page 1. These three levels were the strategy level, the optimizer level and the one dimensional search level. Table 5 lists the levels and the algorithms available. Reference 20, page 5 provides a matrix which

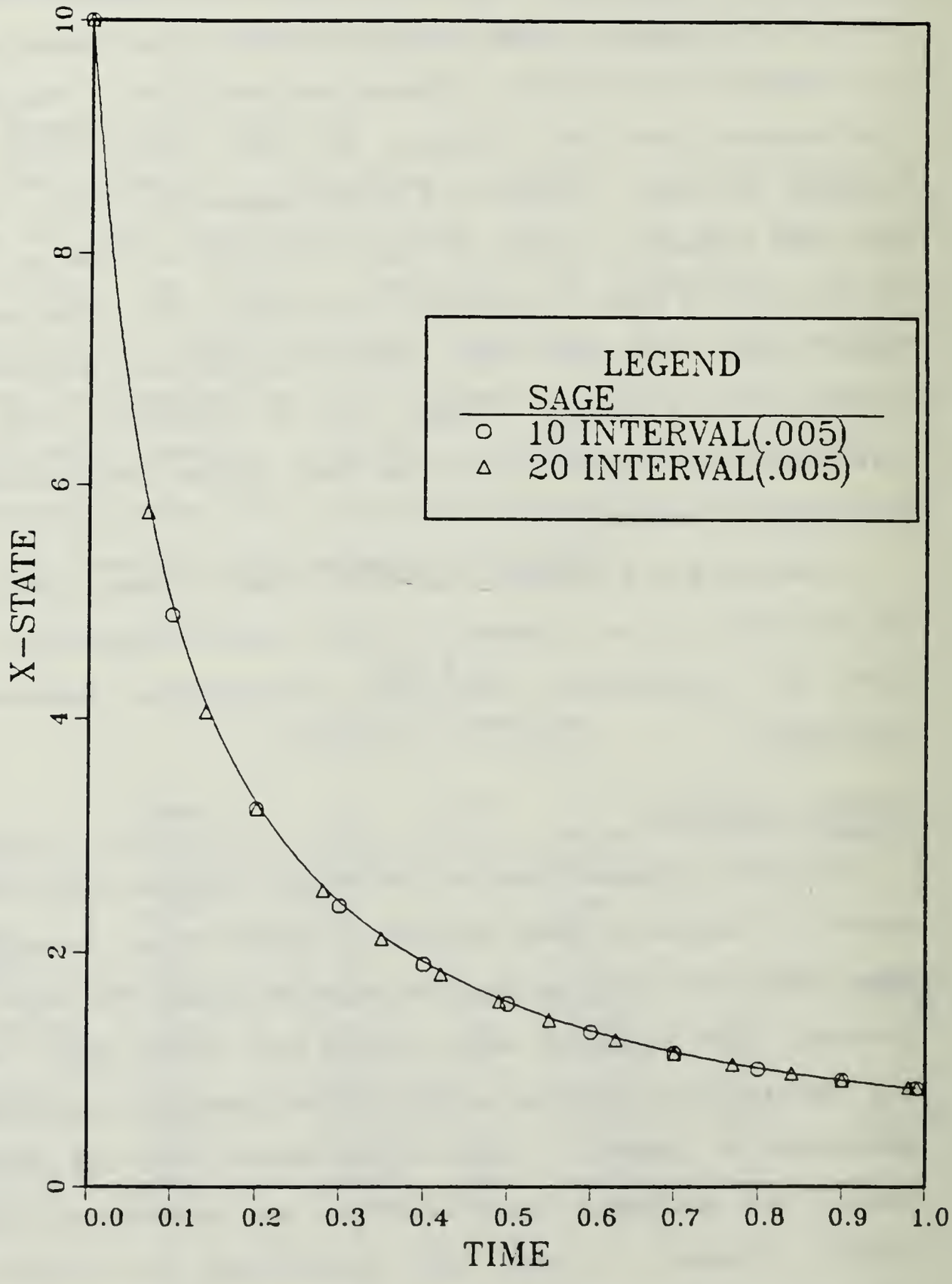


Figure 2.4 X-state Values for Discrete Controls

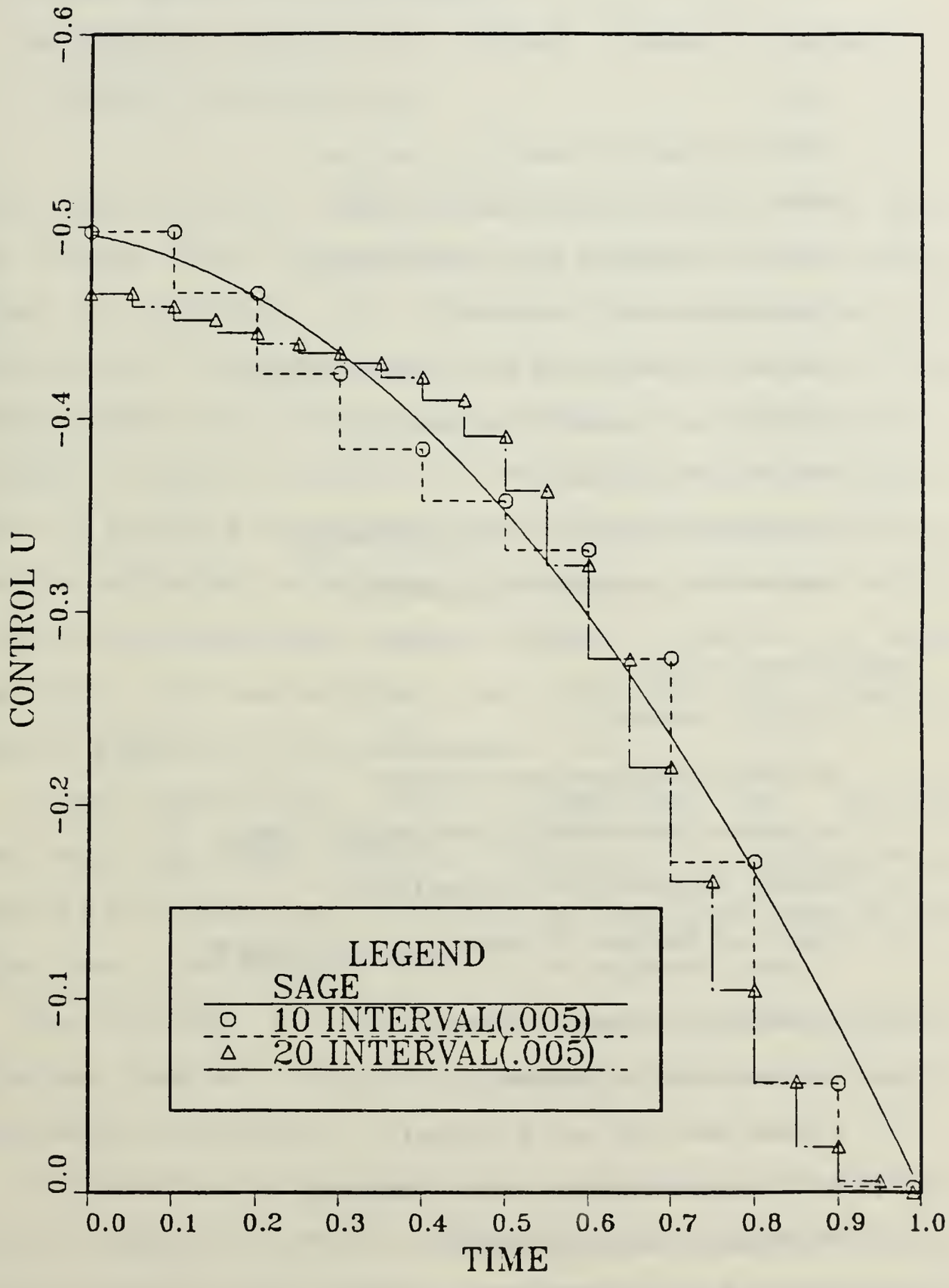


Figure 2.5 Discrete Control Results

TABLE 5

ADS OPTIONS

Strategy (ISRRAT)

0 - None

1 - SUMT, Exterior Penalty Function

2 - SUMT, Linear Extended Interior

3 - SUMT, Quadratic Extended Interior

4 - Cubic Extended Interior

5 - Augmented Lagrange Multiplier Method

6 - Sequential Linear Programming

7 - Method of Centers

8 - Sequential Quadratic Programming

9 - Sequential Convex Programming

Optimizer (IOPT)

1 - Fletcher-Reeves

2 - Davidon-Fletcher-Powell (DFP)

3 - Broydon-Fletcher-Golfarb-Shanno (BFGS)

4 - Method of Feasible Directions

5 - Modified Method of Feasible Directions

One dimensional Search (IONED)

1 - Golden Section Method

2 - Golden Section and Polynomial

3 - Polynomial Interpolation, bounded

4 - Polynomial Extrapolation

5 - Golden Section Method

TABLE 5 (CONTINUED)

6 - Golden Section and Polynomial

7 - Polynomial Interpolation, bounded

8 - Polynomial Extrapolation

gives the meaningful combinations of algorithms to be used for various types of problems. In summary, the two major types of problems are classified as constrained or unconstrained minimization. The Sage problem was unconstrained so strategies 1 through 5, optimizers 1 through 3 and one dimensional search methods 1 through 4 were all possible selections. For a constrained problem, as for the AUV model, strategies 6 through 9, optimizers 4 and 5 and one dimensional search methods 5 through 8 were possible. If the strategy is eliminated, the solution process starts with the optimizer.

Olson [Ref.21:pp. 27-32] summarized the options available and made further reference to Vanderplaats [Ref.22] for additional details concerning these methods and algorithms. The material will not be repeated here.

The strategy, as stated earlier, was optional for all problems; however, an optimizer and one dimensional search algorithm were required.

In determining the combination of strategy, optimizer and one dimensional search method that was the most accurate for solving the Sage unconstrained problem, all the possible

combinations were run. It was quickly determined that the strategy option when invoked caused the execution time and the number of calls to ADS to increase. The solutions were not as accurate as those without a strategy invoked. It was determined that the best optimizer proved to be the Fletcher-Reeves conjugate direction method (IOPT = 1). This method was a simple modification to the first-order method of steepest descent. It involved gradient information normally supplied by using finite difference computations [Ref.22:p. 88]. The conjugate direction method approach picked search directions that were conjugate by definition of the search direction equation:

$$\underset{\sim}{X}^1 = \underset{\sim}{X}^0 - \alpha \nabla F(\underset{\sim}{X}^0) \qquad (2\text{-}3)$$

where:

$\underset{\sim}{X}^1$ = new design variable vector

$\underset{\sim}{X}^0$ = previous design variable vector

α = scalar parameter

$\nabla F(\underset{\sim}{X}^0)$ = gradient of $F(\underset{\sim}{X}^0)$. [Ref. 22:p. 74]

The one dimensional search algorithm that provided the best results with the optimizer was the method of polynomial extrapolation (IONED = 4). The graphical results of Figures 2.4 and 2.5 were those using the above optimizer and one dimensional search combination.

III. PATH OPTIMIZATION FOR A LINEAR MIMO PLANT

A. MIMO PLANT FORMULATION

With the optimization method tested on a SISO unconstrained problem, and realizing computation times were satisfactory for a near real time solution, the method was applied to a linear MIMO plant. Specifically, a linearized model of an under-water vehicle restricted to dive plane motion was developed. The development followed the procedures specified in [Ref. 23: p. 476] with the assumptions that:

i) the vehicle equilibrium condition was along a straight line path;

ii) it traveled at constant speed;

iii) it moved in a fixed horizontal direction, and

iv) it had xg = 0. (xg is the distance the body fixed coordinate system is from the actual center of gravity of the vehicle in the x-direction.)

The standard six-degree-of-freedom dynamic equations of motion of a submarine developed by Gertler and Hagen [Ref.24] and revised by Feldman [Ref.25] were the nonlinear equations that were linearized. Due to the above assumption for motion restricted to the vertical plane, the axial force equation (X) was decoupled from the pitch moment equation (M) and the normal force equation (Z). The equations simplify to the following:

$$-Z'_{\dot{w}}\dot{w}'+(m'-Z'_{w})\dot{w}'-(Z'_{q}+m')q'-Z'_{\dot{q}}\dot{q}' = Z'_{ds}DS'+Z'_{dB}DB' \qquad (3\text{-}1)$$

$$-M'_{\dot{w}}\dot{w}'-M'_{w}w'-M'_{\dot{q}}\dot{q}'+(I'_{y}-M'_{\dot{q}})\dot{q}'-M'_{\theta}\theta'=M'_{ds}DS'+M'_{dB}DB' \qquad (3\text{-}2)$$

The additional coupling equation linking the forward motion speed to the problem is:

$$z' = w' - u'_{0}\theta \qquad (3\text{-}3)$$

where u'_{o} is a constant forward speed. The model is completely nondimensionalized by the vehicle's length of approximately 17 feet, and a velocity of six feet per second. The equations were modelled in state space form with the four states being:

thetadot = rate of change of pitch

theta = pitch angle in radians

w = rate of change of depth

z = depth (positive going down).

A DSL simulation program (STSPACE3 DSL in Appendix B) verified that the model was working properly. The coupling of ADS and DSL followed with path optimization as the goal.

The vehicle from which this model is derived has coupled stern and bow planes. That is, given a stern plane deflection, the bow planes deflect opposite in direction with a given proportionality constant. This aids in maneuvering the vehicle. With this coupling, the actual

control matrix was a scalar, associated with the stern plane deflection only. Controllability was checked and the plant was verified as controllable. However, in the following discussion, the planes were decoupled so that two separate control inputs were optimized over the specified length of time to perform any dive maneuver. This was done for additional controllability.

B. DEFINING THE PROBLEM

When an underwater vehicle travels from one position to another, the path upon which it traverses is a smooth continuous path that does not have discontinuities. If the vehicle is stable, the trajectory is predictable and it may be expected that a path planner using optimization techniques could find the "best" path from the start position to the end. Further, while finding the best path, all stationary obstacles must be avoided, if feasible.

The sample problem was formulated for a maneuver in the vertical plane, a one unit depth change (17.425 feet) in 20 seconds (t' = 7.0), with the vehicle at an initial forward velocity of six feet per second (one unit of nondimensional velocity), without obstacles. The initial conditions were associated with the vehicle traveling at constant horizontal motion at an arbitrary depth (zero in the figures). At time equal to zero (t' = 0.0) the coordinate system was positioned at the vehicles center of gravity. The final state was to occur at time equal to 20 seconds (t' = 7.0)

with Z = 17.425, theta = 0.0, and the control surfaces at their equilibrium positions. In other words, the vehicle was to be at the desired end depth at zero pitch and with control surfaces at the neutral position so that immediately after 20 seconds the vehicle would continue to travel along the desired depth. Before discussing the optimization techniques required to solve such a problem as depth change, the objective function and approach taken will be discussed.

C. THE OBJECTIVE FUNCTION

For the linear plant model given by a system of equations in the form

$$\dot{\underset{\sim}{x}} = \underset{\sim}{A}\,\underset{\sim}{x} + \underset{\sim}{B}\,\underset{\sim}{u}\,, \tag{3-4}$$

the desired function to minimize is an error function of the form

$$\int_0^T [\underset{\sim}{x}_f(t) - \underset{\sim}{x}(t)]^T \underset{\sim}{Q} [\underset{\sim}{x}_f(t) - \underset{\sim}{x}(t)]dt \tag{3-5}$$

In addition, since power consumption is paramount when dealing with self contained underwater vehicles, a minimization of the control energy should be included. The following equation forms a power penalty on the control vector $\underset{\sim}{u}$:

$$\int_0^T [\underset{\sim}{u}(t)\ \ \underset{\sim}{R}\ \ \underset{\sim}{u}(t)]dt \tag{3-6}$$

So, the combined objective function that should be minimized is:

$$\int_0^T \{[\underset{\sim}{x}_f(t)-\underset{\sim}{x}(t)]^T \underset{\sim}{Q}[\underset{\sim}{x}_f(t)-\underset{\sim}{x}(t)] + \underset{\sim}{u}(t)^T \underset{\sim\sim}{R}\underset{\sim}{u}(t)\}dt \qquad (3\text{-}7)$$

This is the quadratic performance index presented by Ogata. [Ref.26:p.753]

For a depth change maneuver all states except the position states should be zero at the end state. In a strict X-Z plane maneuver with the X state decoupled, the final Z position will have a value other than zero (say, z_f); hence, the "f" subscripts in the equation above.

D. THE APPROACH

Four combinations of end time treatments are thus possible to solve this two point boundary value problem:

1) Minimize the quadratic performance index by gain adjustment without end constraints.

2) Use the ADS defined end constraints and minimize the objective.

3) Combine ADS defined end constraints with penalty functions added to the objective.

4) Minimize the objective combined with penalty functions on the desired terminal conditions, without ADS end constraints.

The results of each combination is presented in the following discussion and graphically in the accompanying figures. A discussion of the accuracy and timing will follow at the end of the chapter. All methods required the

use of side constraints to limit the bow and stern plane deflections.

1. Side Constraints

After considering the actual movement of an underwater vehicle through the water, it was realized that several restrictions needed to be placed on the vehicles' maneuvering surfaces. The maximum stern plane deflection was limited to no more than a 10 degrees up or down angle from the equilibrium position, and the bow planes were limited to approximately 15 degrees up and down angle. To effect such a restriction in ADS, the side constraint approach must be invoked. In Vanderplaats' formulation of the problem for the ADS package, the side constraints are the upper and lower bounds of the design variables to be optimized [Ref.20]. They are identified as VLB, VUB respectively in the programs. This is an advantage to a constrained optimization problem because it limits the search area in which to minimize the objective function. This kind of constraint can be bumped up against and become active, which means that the constraint value is in effect for a specific design variable. If all of the side constraints are active, then there is a good possibility that the true minimum has not been reached as the design variables have most likely hit the constraint before the gradients had zeroed (found their minimum). If the side

constraints are active then the problem may need to be redefined within the limits of the side constraints.

2. Quadratic Performance Index Weighting Alone

The quadratic performance index (Eq. 3-7) was the objective function that ADS minimized using the unconstrained minimization techinque concluded from the previous work in Chapter II. The method, repeated here for the reader, was the Fletcher-Reeves optimizer (IOPT = 1) and the Polynomial extrapolation one dimensional search method (IONED = 4).

The run times were favorable at less than four seconds for the weighting matrix determined "best." It was observed that as weighting was added to improve the depth accuracy the pitch accuracy worsened, and vice versa. As a result of this, the weighting matrix that was determined to produce the "best" result was the identity matrix. The "best" result produced a five percent difference between the desired depth and the actual depth. The pitch angle, however, ended up short by over six degrees although it was falling off toward zero from its maximum deflection.

The results indicate, and are verified by another deeper depth change for the same terminal time, that although depth can be achieved relatively accurately the additional desired end state of zero pitch angle was not achievable. Figures 3.1 and 3.2 present the results of two different depth changes using the same gain matrix

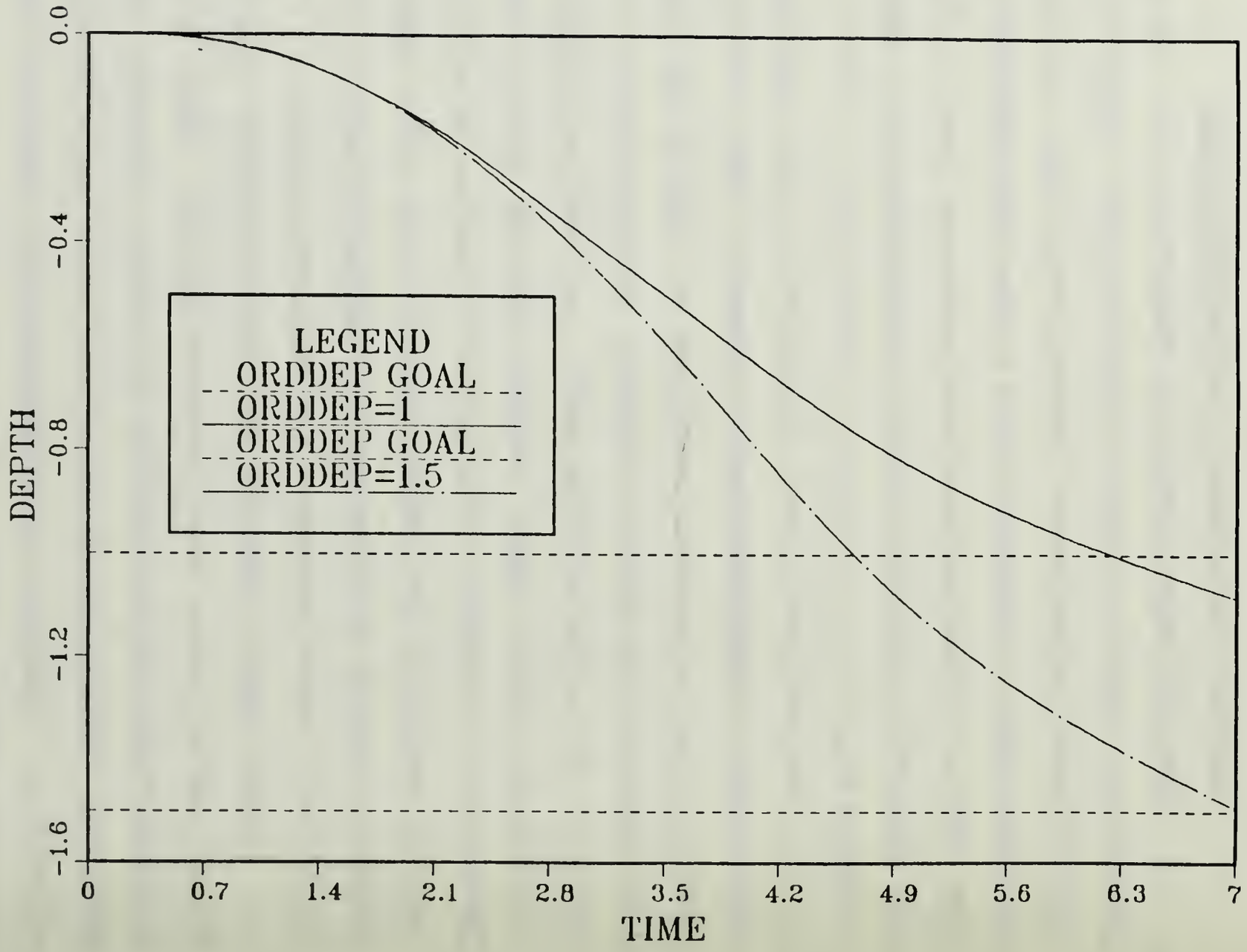


Figure 3.1 Linear Depth Maneuvers using Objective Function Minimization

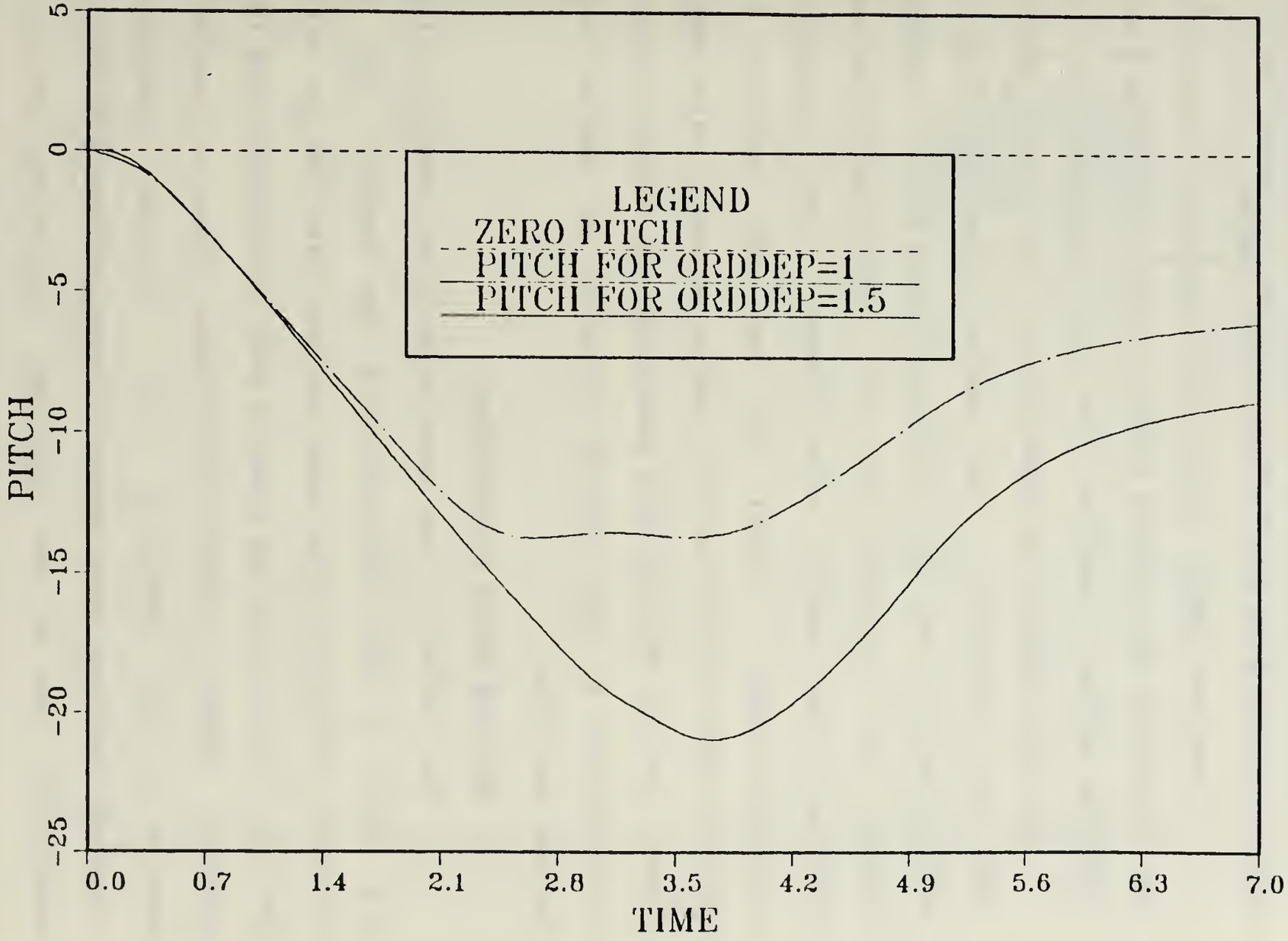


Figure 3.2 Pitch Plot for Linear Depth Maneuvers

determined for the first depth change (the deeper depth). The accuracy of the terminal depth, in general, increased with an increase in size of the depth change.

The same depth change was then given twice as much time to get to the desired depth to test the adequacy of the weighting matrix to varying dive durations. Figures 3.3 and 3.4 show the results of these runs. The figures show that depth was increasing at the terminal time and pitch angle was falling off, but at an extremely slow rate. Figures 3.1 through 3.4 illustrate that the "best" weighting was a function of the specific dive ordered, in the unconstrained end time cases. Clearly the two point boundary value problem could not be strictly enforced using this approach alone. It was obvious that something was needed to "drive" the solution to the desired terminal end states at the desired end time.

3. ADS End State Constraints

The choice of end constraint types available in ADS are found in the definition of the vector, IDG. IDG contains parameters that must be specified in the call to ADS and it specifies the type of each constraint used in the problem. Table 6 lists the available types of constraints handled by ADS [Ref.20:p. 11]. Further investigation revealed that the best method to satisfy the end condition constraints was to use two ADS constraints defined as nonlinear equality types (IDG = -1) for each terminal end

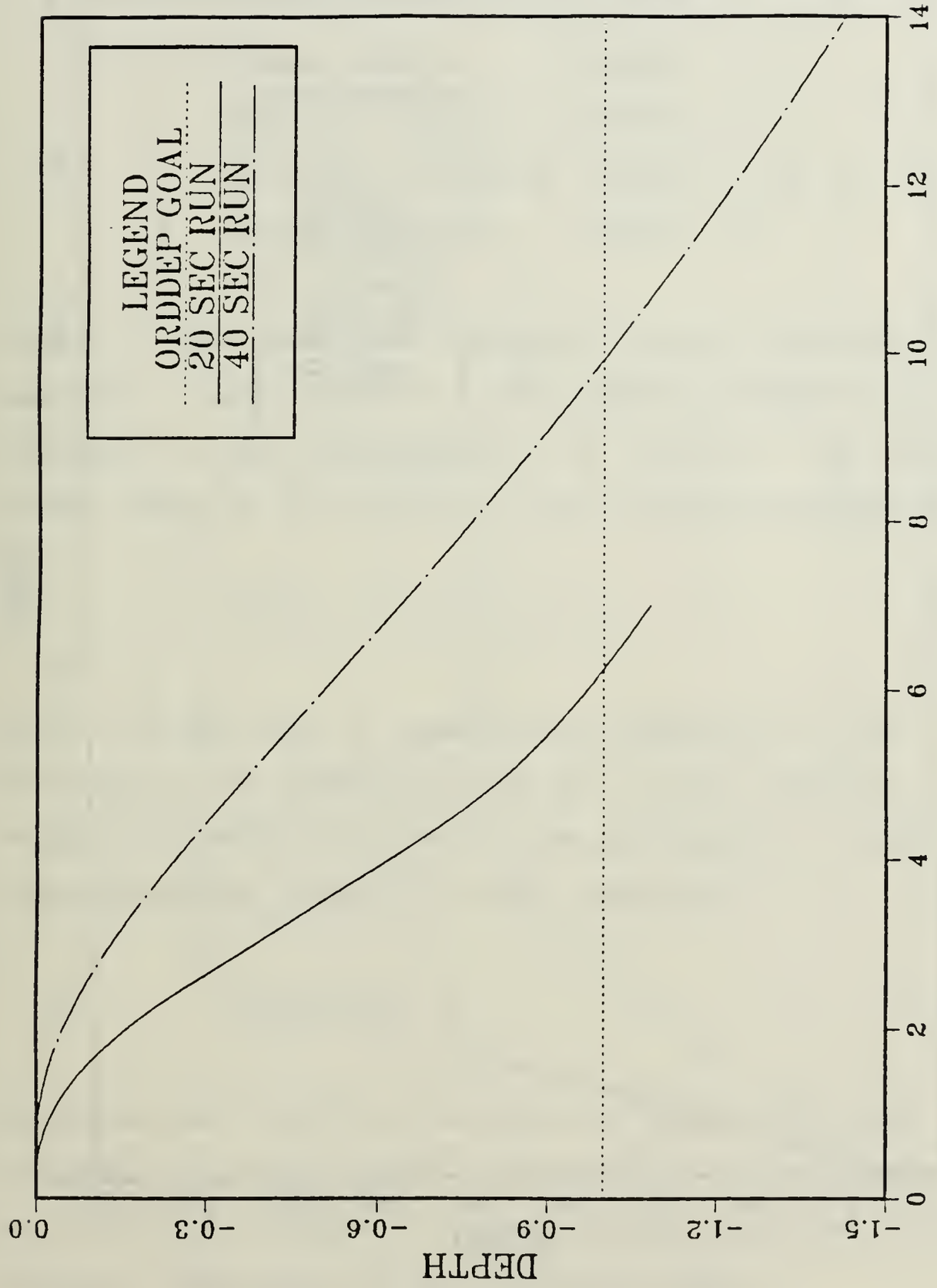


Figure 3.3 Effect of Increased Time on Ability to Achieve Desired Depth

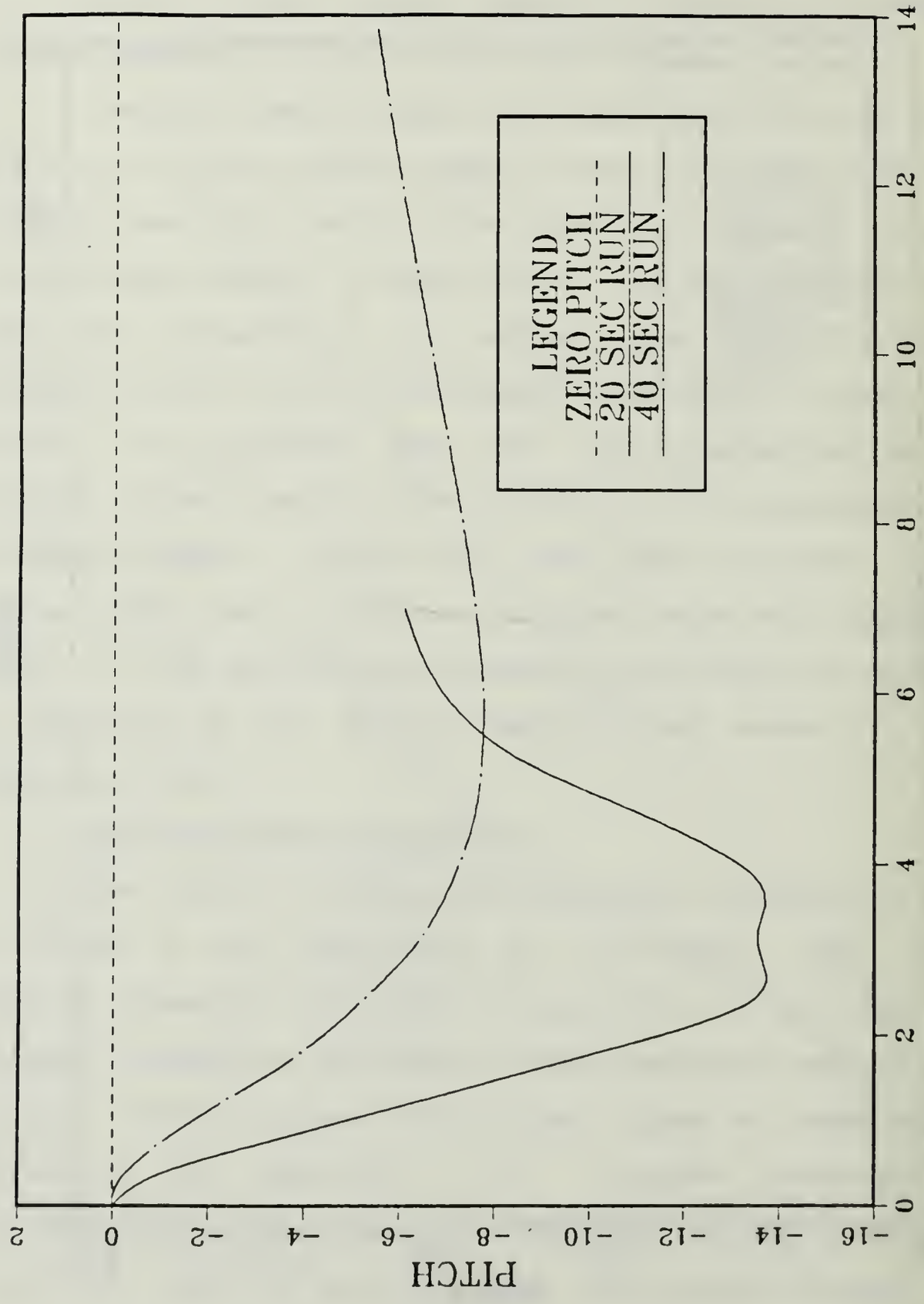


Figure 3.4 Pitch Plot Corresponding to Maneuver in Figure 3.3

TABLE 6

CONSTRAINT TYPES IN ADS

1. Linear equality (IDG(I) = -2)
2. Nonlinear equality (IDG(I) = -1)
3. Nonlinear inequality (IDG(I) = 0 or 1)
4. Linear inequality (IDG(I) = 2)

condition specified; the constraint and its negative. For example, a depth maneuver to an arbitrary depth of 50 feet required two end constraints to be written. ADS requires one of these to be entered via the following inequality:

$$G(1) \leq Z - 50. \tag{3-8}$$

Here, ADS was told to recognize the equality via IDG = -1. Thus when z is greater than 50 feet, this condition is not satisfied and it does penalize the optimization. Similarly, introducing the negative of this constraint,

$$G(2) \leq 50 - Z \tag{3-9}$$

enforces that Z must be less than or equal to 50 feet. ADS attempts to satisfy both constraints on Z by forcing the value of Z to 50 feet. A similar method was used to apply terminal constraints on all the final states.

In addition to the constraint relationships, the optimization algorithm had to be changed in order to carry out a constrained minimization problem. It was necessary to conduct a study of the effects on accuracy and run time, as well as a study on which optimization algorithm was best suited to this problem.

The ADS choices for such a problem as this were (refer to Table 5) the 047, 057, 657, 533, and 133 options. Table 7 presents the results of running this ADS constrained terminal condition problem using each of the five options. By invoking the use of a strategy option, the run time increased a minimum of 66 percent over the shortest runtime. The "best" method, considering both accuracy and runtime, was the 057 option. This was used in all the remaining ADS constrained cases.

Solving the terminal constrained problem using the 057 option did require some weighting of the constraints in order to produce the "best" accuracy and run time. The weighting of the constraints was determined to be 0.5 for the two depth constraints and 10.0 for the two pitch constraints. When these weights were applied to a deeper depth maneuver, the final depth was approximately six inches from the desired depth. The pitch angle was also small, less than 0.02 degrees. Figures 3.5 and 3.6 present the results graphically for both depth and pitch, and for both depth changes. The results illustrate the versatility

TABLE 7

ADS OPTIONS RESULTS

Option	047	057	657	533	133
depth	-.99595	-1.0002	-1.0000	-.99637	-.99717
Pitch	3.4×10^{-16}	-.0231	-5.5×10^{-10}	6.4×10^{-3}	-.0187
Pitch rate	3.29×10^{-2}	2.93×10^{-2}	2.93×10^{-2}	1.96×10^{-3}	1.92×10^{-2}
depth rate	5.08×10^{-3}	3.27×10^{-3}	4.60×10^{-3}	3.6×10^{-3}	3.44×10^{-3}
Reruns	192	112	191	517	535
Run time (sec)	4.21	3.16	4.94	8.60	8.86
Obj	1.6577	1.7421	1.7012	1.5669	1.5671

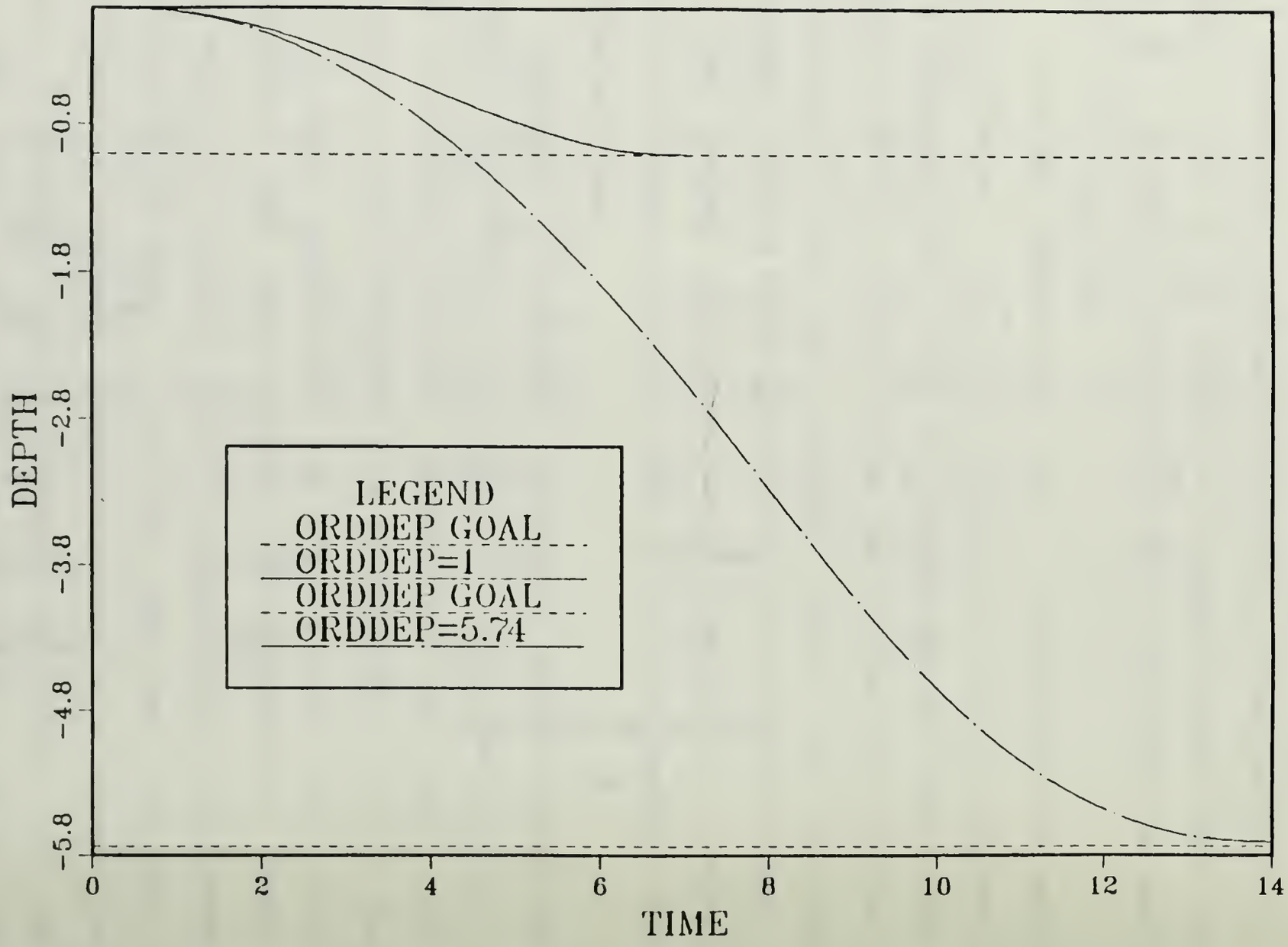


Figure 3.5 Depth Results Using ADS Constraints on Both Depth and Pitch

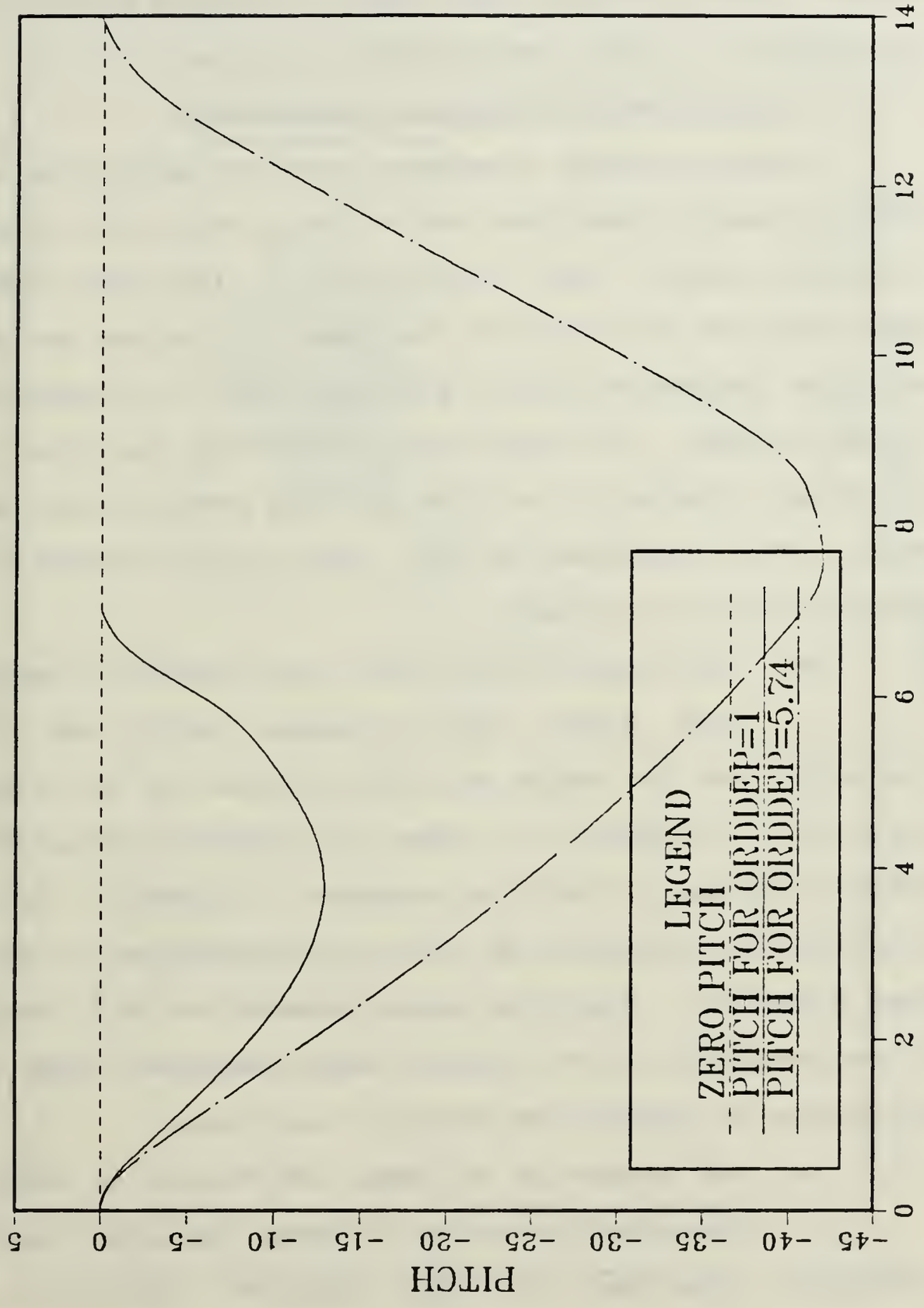


Figure 3.6 Pitch Plot for Maneuvering in Figure 3.5

of the ADS constraints to handle the typical range of depth changes without changing the weighting factors either on the constraints or within the objective function.

4. ADS and Penalty Function Combinations

Another method to enforce terminal conditions is by creating penalty functions and adding them to the original objective function [Ref. 19:p. 315]. Given the terminal conditions for this problem, the penalty function for depth and pitch respectively are, (ORDERED DEPTH - Z) squared and (PITCH) squared. Two cases were considered, the first case was an ADS constraint on pitch and the penalty function on depth. The second was an ADS constraint on depth and a penalty function on pitch.

a. ADS Constraint on Pitch and Penalty on Depth

This first case required both the pitch constraint and the depth penalty function to be weighted close to a thousand in order to achieve satisfactory accuracy for the 17 foot dive maneuver. Figures 3.7 and 3.8 illustrate the trajectories and pitch performance of the two dives discussed. Weighting again appeared to be a function of the magnitude of the depth change maneuver using this combination to enforce the terminal conditions.

b. ADS Constraint on Depth and Penalty on Pitch

Reversing the roles of depth and pitch terminal constraints resulted in much improved accuracies and response then the previously discussed method. With the ADS

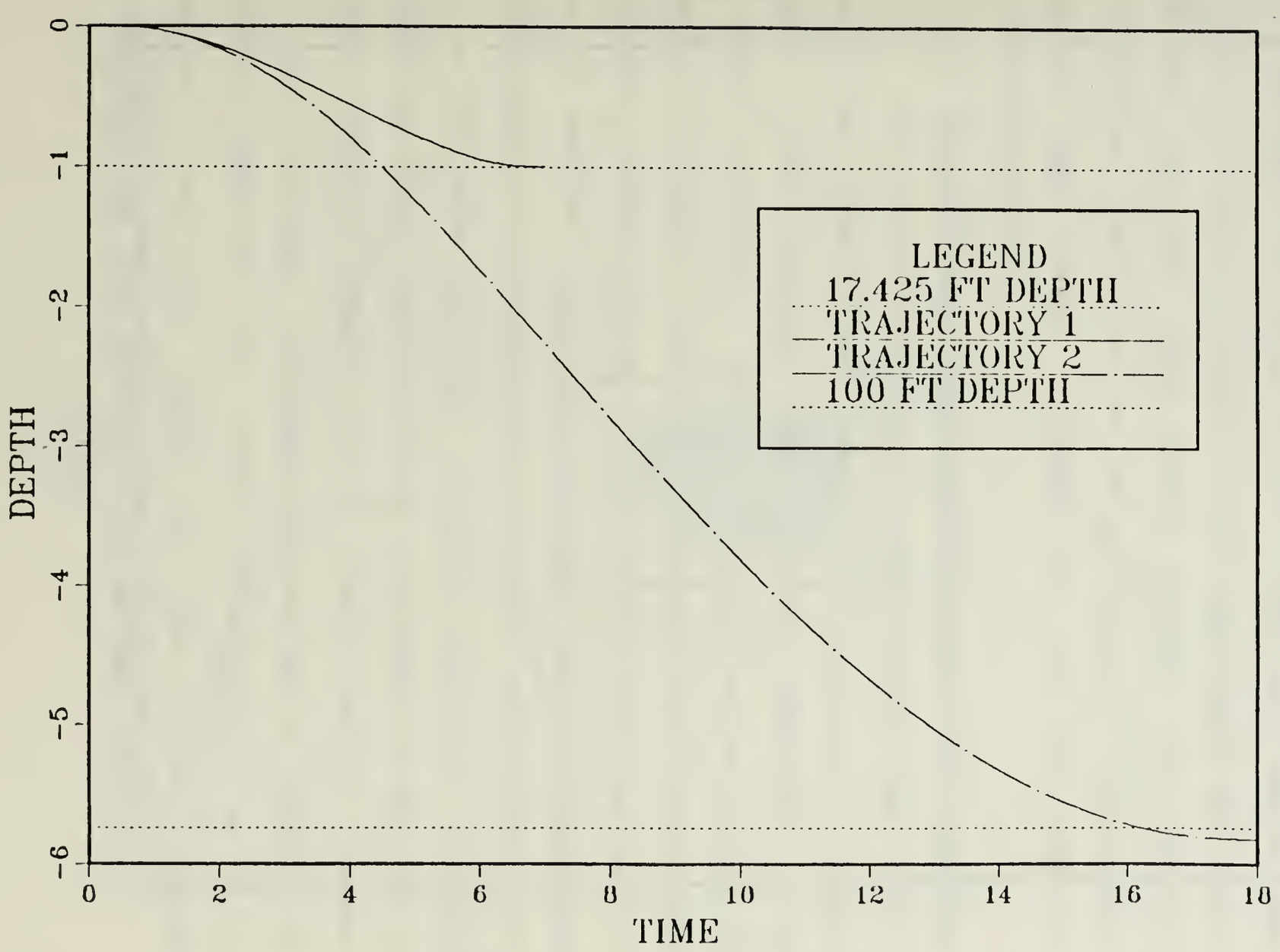


Figure 3.7 Depth Result Using ADS Constraint on Pitch and Penalty on Depth

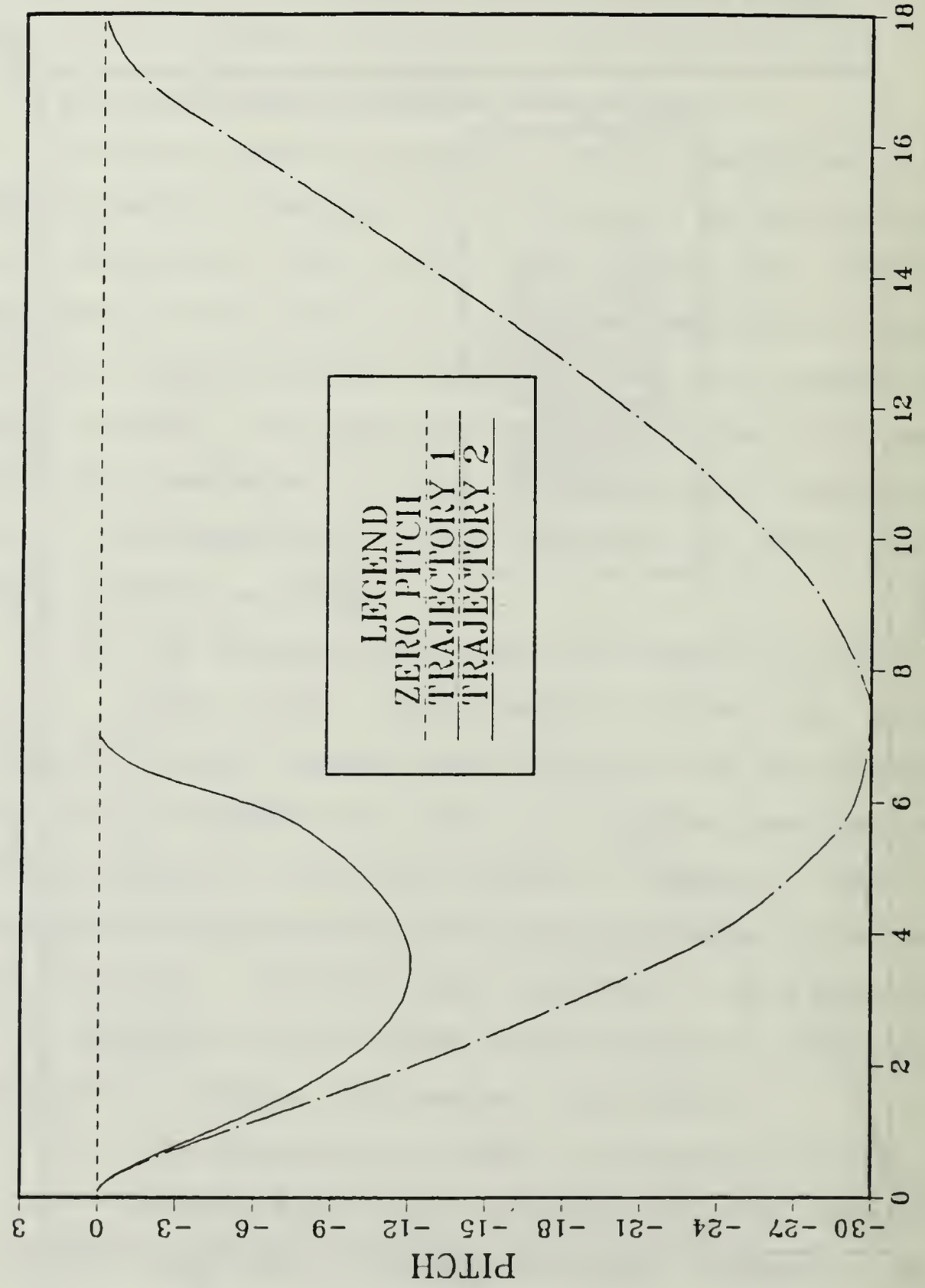


Figure 3.8 Pitch Plot Corresponding to Maneuver in Figure 3.7

depth constraints weighted at unity and the pitch penalty function weighted at 70.0 for the 17 foot dive, the depth was perfectly satisfied and the pitch angle was extremely accurate. The performance of this combination when applied to the 100 feet depth change maneuver was considered acceptable. Weighting was not a function of the magnitude of the depth change. Figures 3.9 and 3.10 graphically present the results.

5. Unconstrained Minimization Using Both Penalties

The final minimization method studied went back to using the unconstrained minimization technique and applied both of the terminal conditions as penalties on the objective function. Depth and pitch terminal conditions were put in penalty functions previously described and added external to the integral of the original objective function. No ADS defined constraints were used and ADS optimization combination 014 was applied as before. This method proved to be ill-fated for this objective function. Increased weighting on the penalties drove the answer to zero and the "best" weighting proved to be the identity matrix. However, at this weighting, the depth achieved was a third of the ordered depth but the pitch was acceptable at 0.0713 degrees. For these reasons the data is not displayed graphically and this method was not further analyzed.

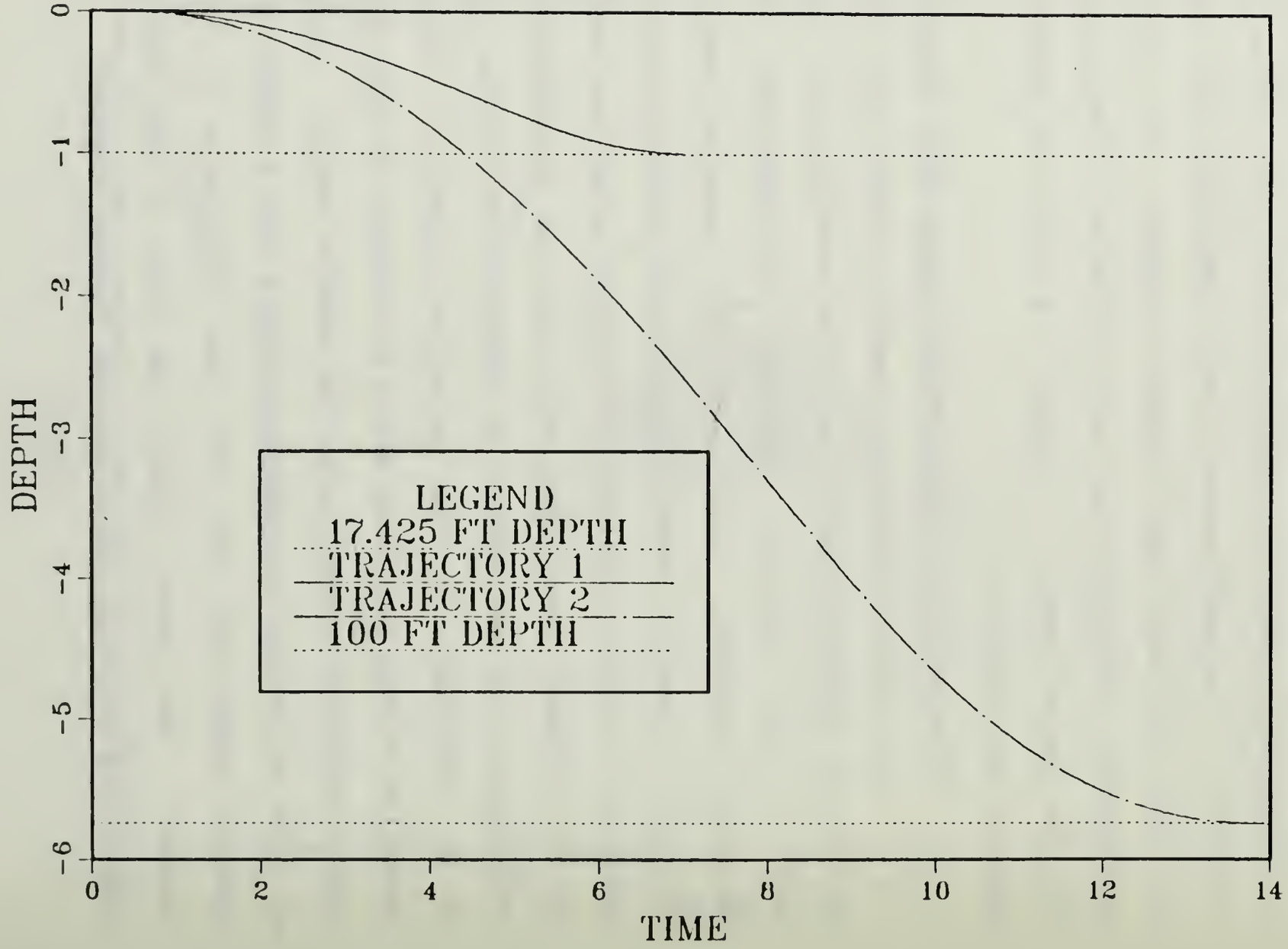


Figure 3.9 Depth Results Using ADS Constraint on Depth and Penalty on Pitch

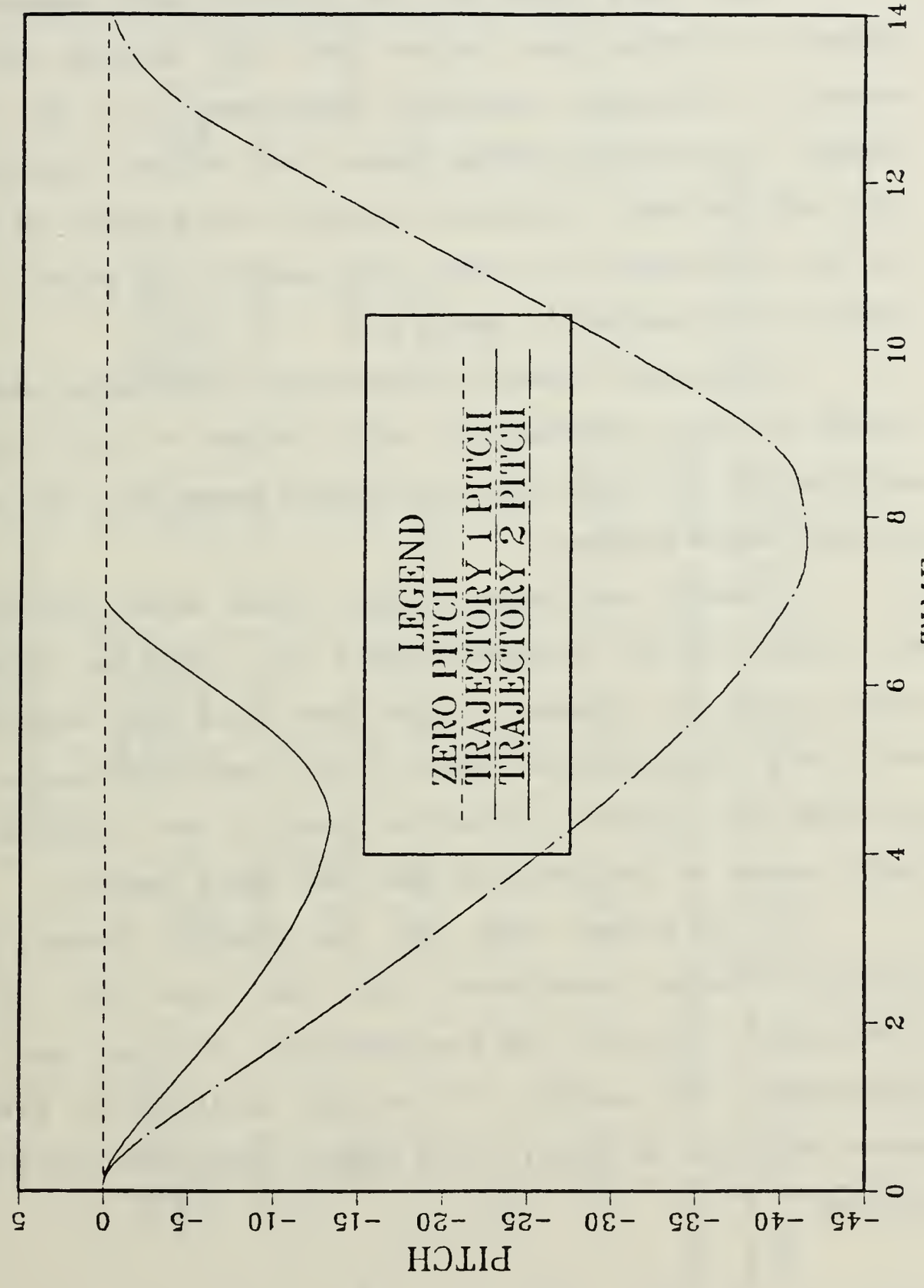


Figure 3.10 Pitch Plot Corresponding to Maneuver in Figure 3.9

6. Timing and Accuracy

Thus, only three of the methods were compared to choose the method best suited for this problem and the obstacle avoidance situation discussed in the next chapter. The three methods were: ADS defined constraints only, ADS defined constraint on pitch and penalty on depth and ADS constraint on depth and penalty on pitch. The results are compiled in Table 8.

The timing criteria together with desirable accuracy caused the "ADS constraints only" method to be the best solution for the path planning problem posed as a two point boundary value problem.

Although the deeper dives' final depth was off by six inches in the selected method and pitch was slightly over a tenth of a degree, it was clear that this method had very good accuracy and near real time performance, it produced the smallest objective function and required the least reruns on the average than the other methods.

It is evident that for the type of dynamic path planning problem undertaken here, the need for an end constraint to "drive" the trajectory to its final goal was determined. The ability of the ADS optimization program proved feasible to satisfy this simple linearized trajectory problem.

TABLE 8

TIME/ACCURACY DATA

	ADS Constraints only (17 ft dive)	100 ft dive	ADS Pitch/ Penalty on Depth (13 ft dive)	100 ft dive	ADS Depth/ Penalty on Pitch	100 ft dive
Depth[1]	-1.0002	-5.7088	-.99917	-5.8223	-1.0000	-5.7517
Pitch (degrees)	- .0234	.11164	.0156	- .022	.0006	- .21830
OBJ	1.7421	97.128	1.7664	110.44	1.8831	100.18
Reruns	112	202	210	141	181	474
Run time (sec)	3.05	7.08	4.50	6.40[2]	4.11	14.83

[1] Depth is non dimensionalized by 17.425 ft (length of vehicle)

[2] Maneuver required an additional 2 nondimensional time units in order to reach the depth of 100 ft that the other two methods did in 14 units

IV. PATH PLANNING APPLICATIONS

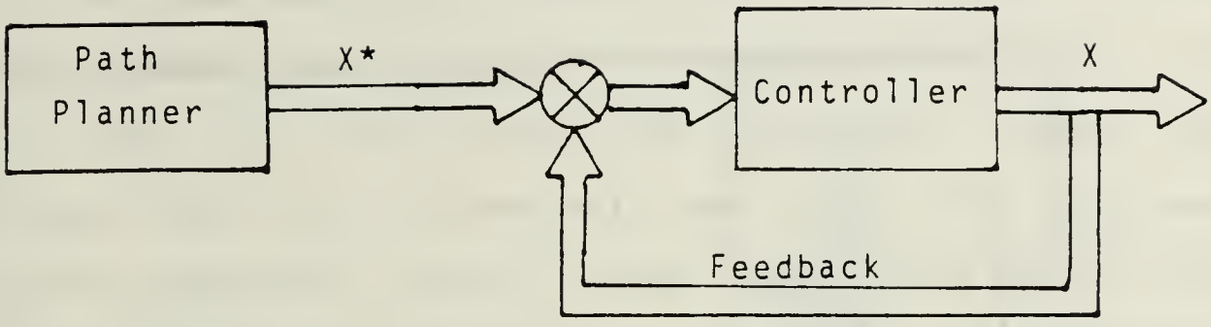
A full scale nonlinear simulation model of the study vehicle was developed by Lt. Boncal in Reference 18. The model was a full six degree-of-freedom model with 12 states that took into account all properties of the vehicle and its motion through water. Comparison of the previous linear model results with those of the nonlinear model were made for the following reasons:

1) the solution to the linear problem would be easier and, therefore, would require less real time to compute, and

2) a determination was necessary as to whether the linearized version would be accurate enough compared to the nonlinear version.

A. LINEAR VERSUS NONLINEAR PATH PLANNING

Each time the optimization program was run, the data that it produced provided the control vector for each control surface used, e.g., the stern plane and bow plane time histories. Also, the time history of the states from beginning to end were computed. It was planned that this state time history would be the desired state trajectory which would be supplied to a state servo for the vehicle. Figure 4.1 illustrates this design concept. The servo then would follow the desired state and keep the error minimal between the actual position and the desired position,



$\underset{\sim}{X}^*$ optimum state vector

Figure 4.1 Path Planner and Controller Relationship

therefore following the optimal path. Consequently, the accuracy of the X history is of critical importance.

With the assumptions used in the simplified linear model,it was obvious that the straight ahead motion (the x position) would not correspond well to the nonlinear model. Since the depth was a function of the horizontal speed, the z-position states could be expected to be in error as well. These results can be seen graphically in Figure 4.2 where both the nonlinear and linear optimized trajectories are represented in the x-z plane. Due to the decoupled axial force equation (X), the linear trajectory extends to the right of the nonlinear trajectory. Due to the nonlinearities in the nonlinear model, the vehicle descends more rapidly, then levels out for a longer period of time. The linear model tends to make a more gradual descent and spend less time leveling out.

The time to achieve the nonlinear optimal trajectory was an order of magnitude greater than for the linearized model

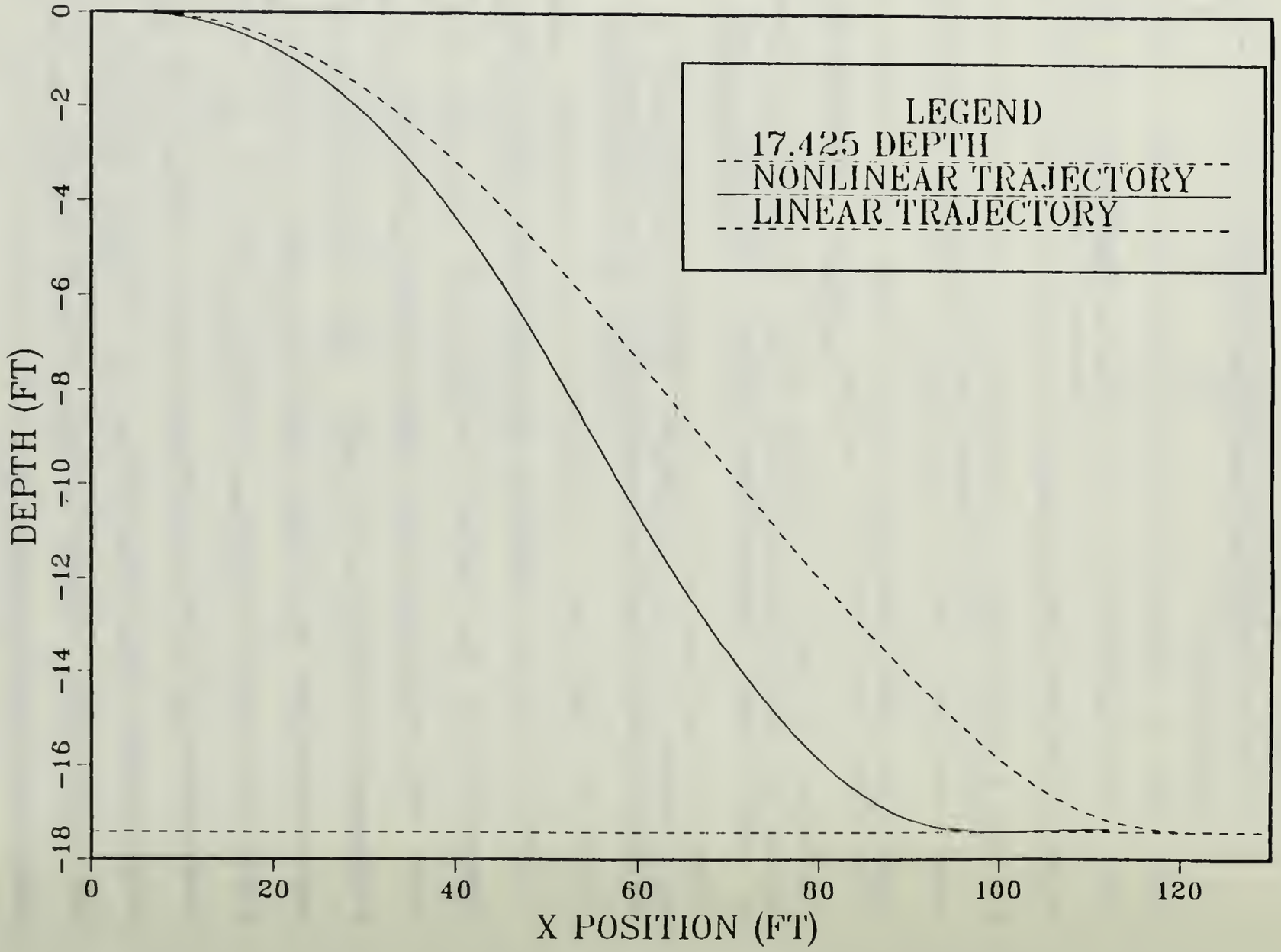


Figure 4.2 Linear and Nonlinear X and Z State Comparison

using the same weighting factors for both models. This greatly hinders the feasibility of utilizing the full non-linear model for real time path planning. Since the nonlinear model was an entirely new plant, the analysis discussed previously for the linear model would have to be applied to improve the timing and accuracy in the nonlinear case. A study was conducted on the effect of weighting the ADS constraints only and an improvement from 60 seconds to approximately 22 seconds was achieved. Additional time may be saved if the best Q matrix could be determined as well.

One additional test was run on the comparison of the two models. As mentioned earlier, the ADSL program also produces the control vector that achieved the optimized states. In this test, the control vector for the linearized model (Fig. 4.3a) was supplied to the nonlinear simulation. The results showed that the nonlinear model descended to within .05 feet of the ordered depth and the pitch was slightly over three tenths of a degree at the final time (Fig. 4.3). Recalling that, although the states do not match up well enough to use the simplified linear model due to the oversimplification of the decoupled axial force equation (Fig. 4.2), the essential dynamics in the linear model were nevertheless valid since the desired depth was achieved using the combination above.

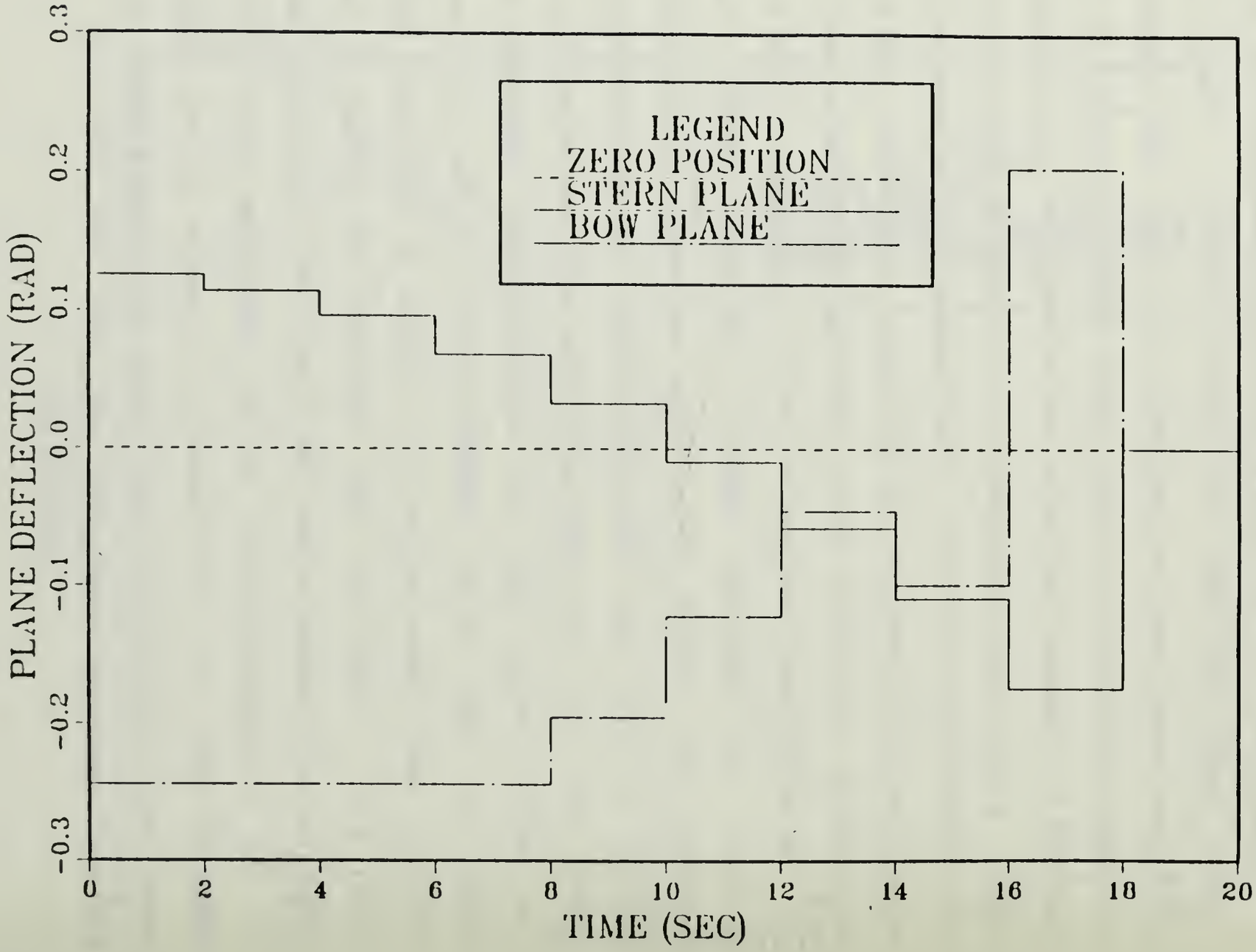


Figure 4.3a Stern and Bow Plane Control

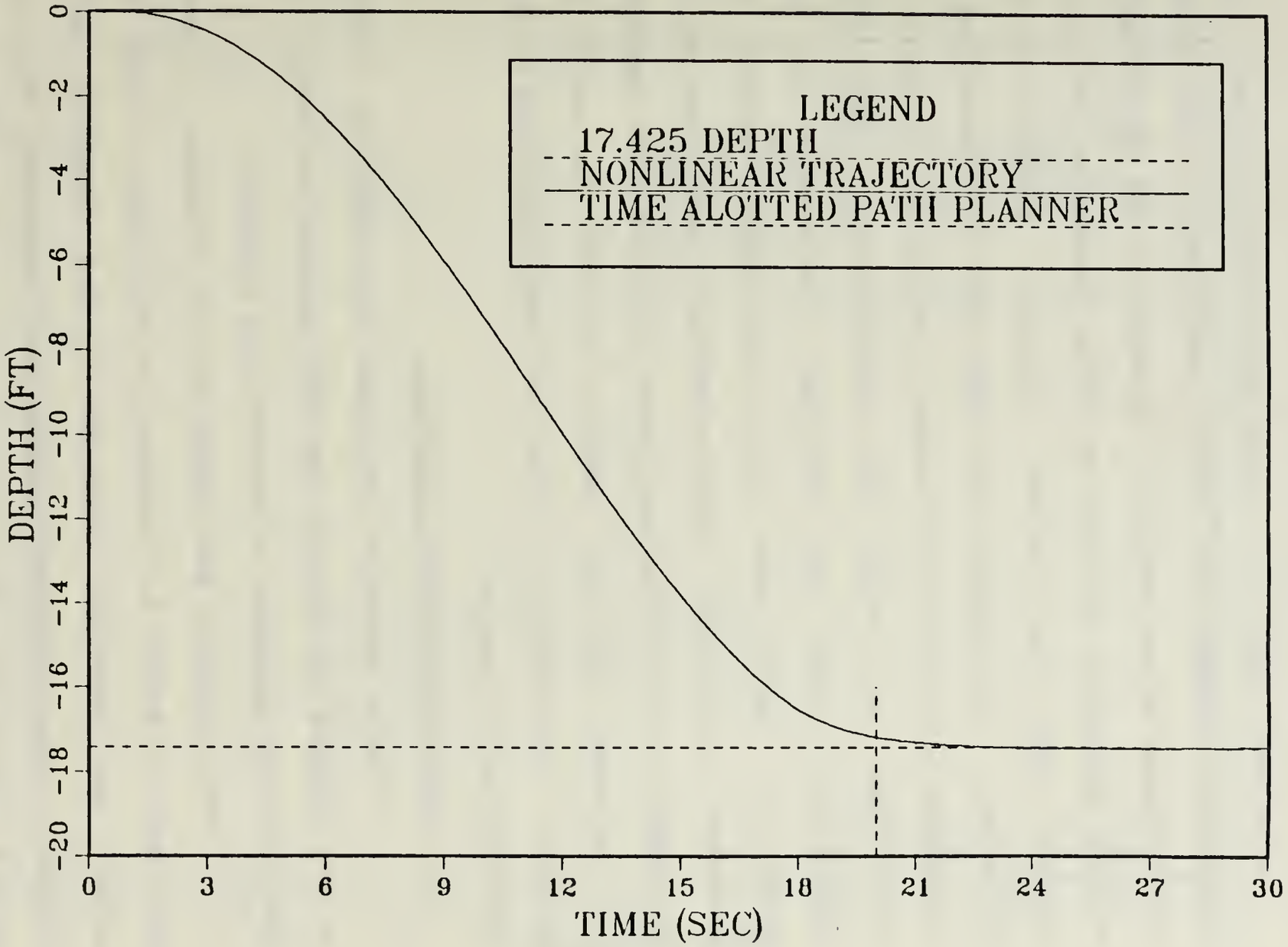


Figure 4.3 Nonlinear Trajectory Using Linear Path Planner Control

B. OBSTACLE AVOIDANCE

A primary goal of a path planner is to plan a safe trajectory between two points specified by the command intelligence of the vehicle. Between the two points may be visible obstacles, as well as hidden obstacles that may be unknown until the vehicle commences further on its trajectory. Consequently, the safe path must sometimes be recomputed in real time to avoid a collision. Figure 4.4 illustrates this with two trajectories. The first trajectory (called the "nonlinear trajectory") was the optimal or "best" path to the new depth without any obstacles in the path. With an obstacle positioned at the (+) symbol, the obstacle avoidance algorithm must compute a new trajectory for the vehicle to take to avoid the obstacle. The obstacle in this study was considered to be a fixed obstacle in space.

1. Obstacle Avoidance Algorithm

A simple obstacle avoidance algorithm was written for the purpose of testing the ADSL program to ensure its capability was adequate to perform this function.

The algorithm was based on the procedure a human would use to avoid an obstacle in his path. Once the obstacle was detected, interest in the obstacle would heighten. The distance between the individual and the obstacle would be kept track of, and the human would take appropriate action to avoid, or make the decision to

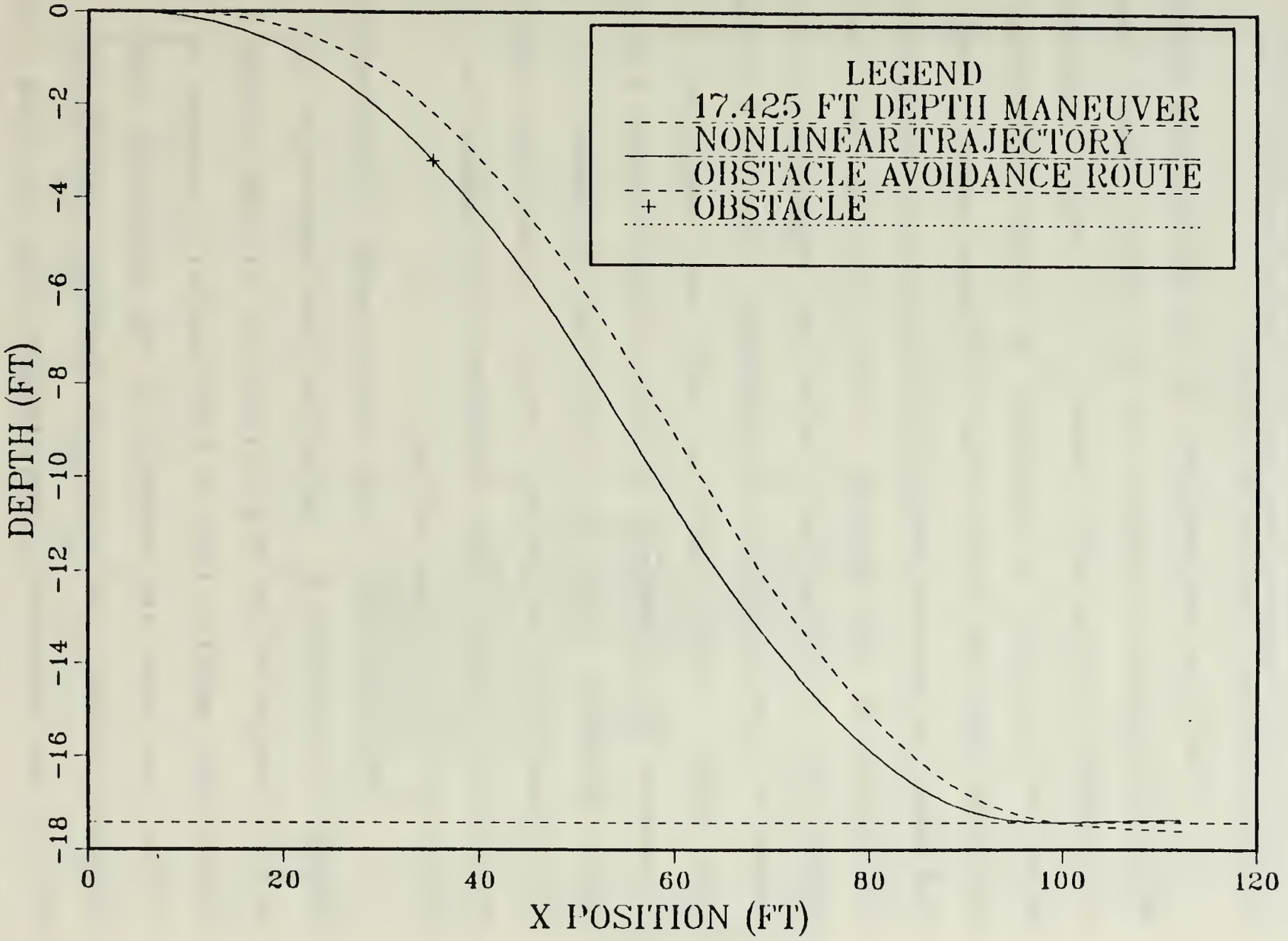


Figure 4.4 Obstacle Avoidance Trajectory

collide, and pay the penalty. Once past, interest in the object would decrease to zero. In an AUV, collision with the obstacle will not be an option.

The size and position of any obstacle was determined to be information supplied by the command intelligence of the vehicle to the path planner. The command intelligence also was assumed to generate an area surrounding the obstacle that was termed the "avoidance zone." This zone was based on the size and maneuverability of the vehicle and the threat of the obstacle. This zone was what the path planner saw as the obstacle which the trajectory of the AUV center of gravity could not penetrate. For the purposes of this study, this obstacle data was emulated by the programmer who picked an arbitrary radius around a selected point obstacle. This is illustrated in Figure 4.5. Also for purposes of the present study, a one foot radius was arbitrarily chosen. The avoidance zone appears oval in the figure due to the axis scaling.

The algorithm, which can be seen in the THE1ND DSL program in Appendix B, takes the time from start until the vehicle would be at the x-position of the obstacle and divides that equally into 10 positions with respect to time. As the positions are reached at the specified times, the distance between the vehicles' current position and the obstacle was computed. The distance then was made into an inequality constraint using ADS and defining IDG = 0 (refer

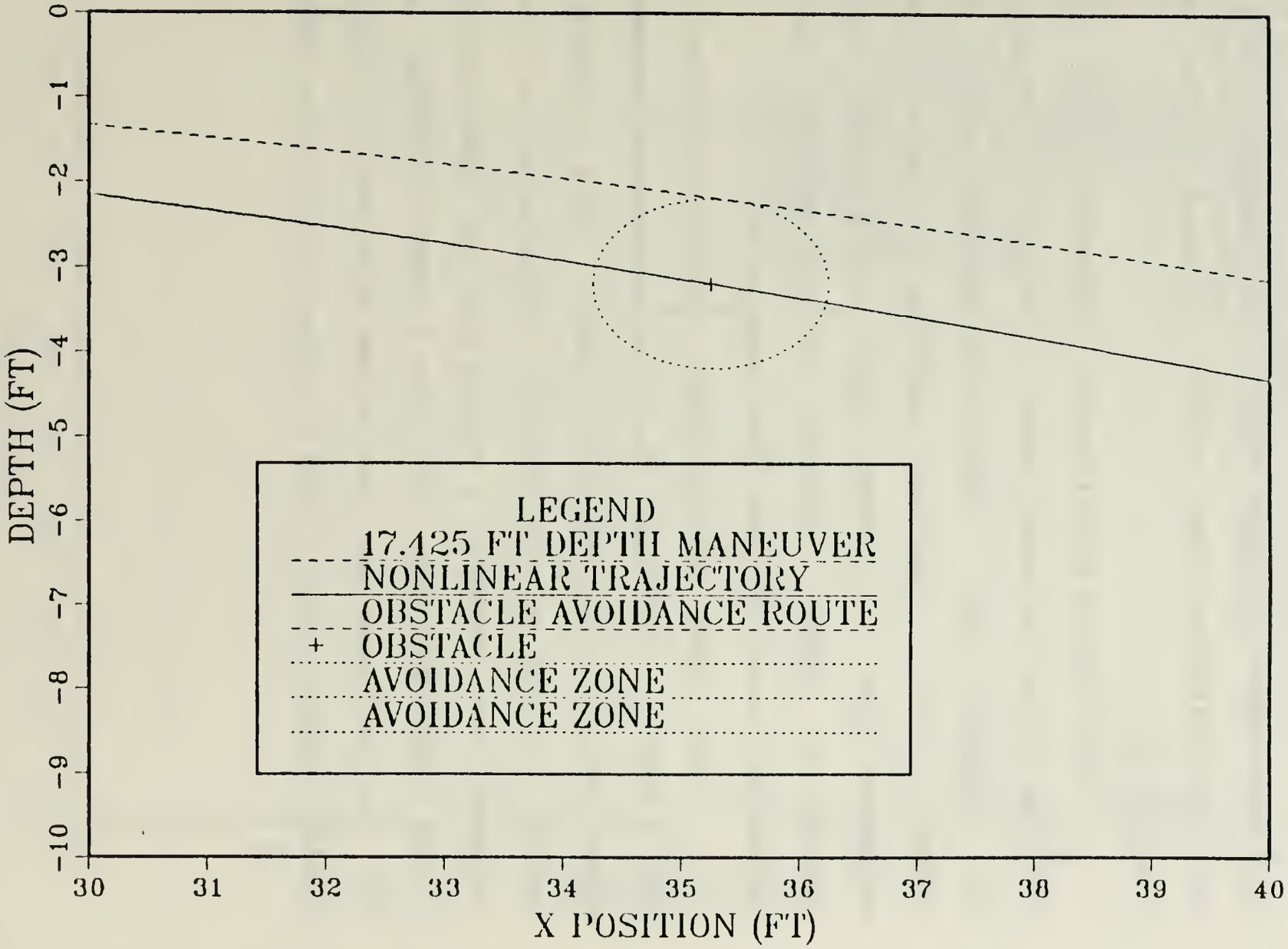


Figure 4.5 Avoidance Zone

to Table 6). Each of these ten new constraints had to be satisfied by ADS. Figure 4.6 illustrates the results using the full scale nonlinear model. The obstacle was avoided and the avoidance zone was not violated by the vehicle trajectory.

This simple algorithm only concerned itself with the obstacle up to the point where it was at the top center of the avoidance zone. This point may not always be the closest point of approach depending on the size of the avoidance zone. In these cases, the algorithm could be changed to compute distances to the obstacle for some designated amount of time after passing it.

The linear and nonlinear obstacle avoidance runs were timed to determine the time difference between the obstructed and unobstructed run times. Table 9 presents the results. It can be seen that the linear model run time was increased four tenths of a second from the unobstructed case to the obstructed case. This has a great advantage, again, over the nonlinear case due to the fact that the nonlinear problem doubled its run time in order to avoid the obstacle.

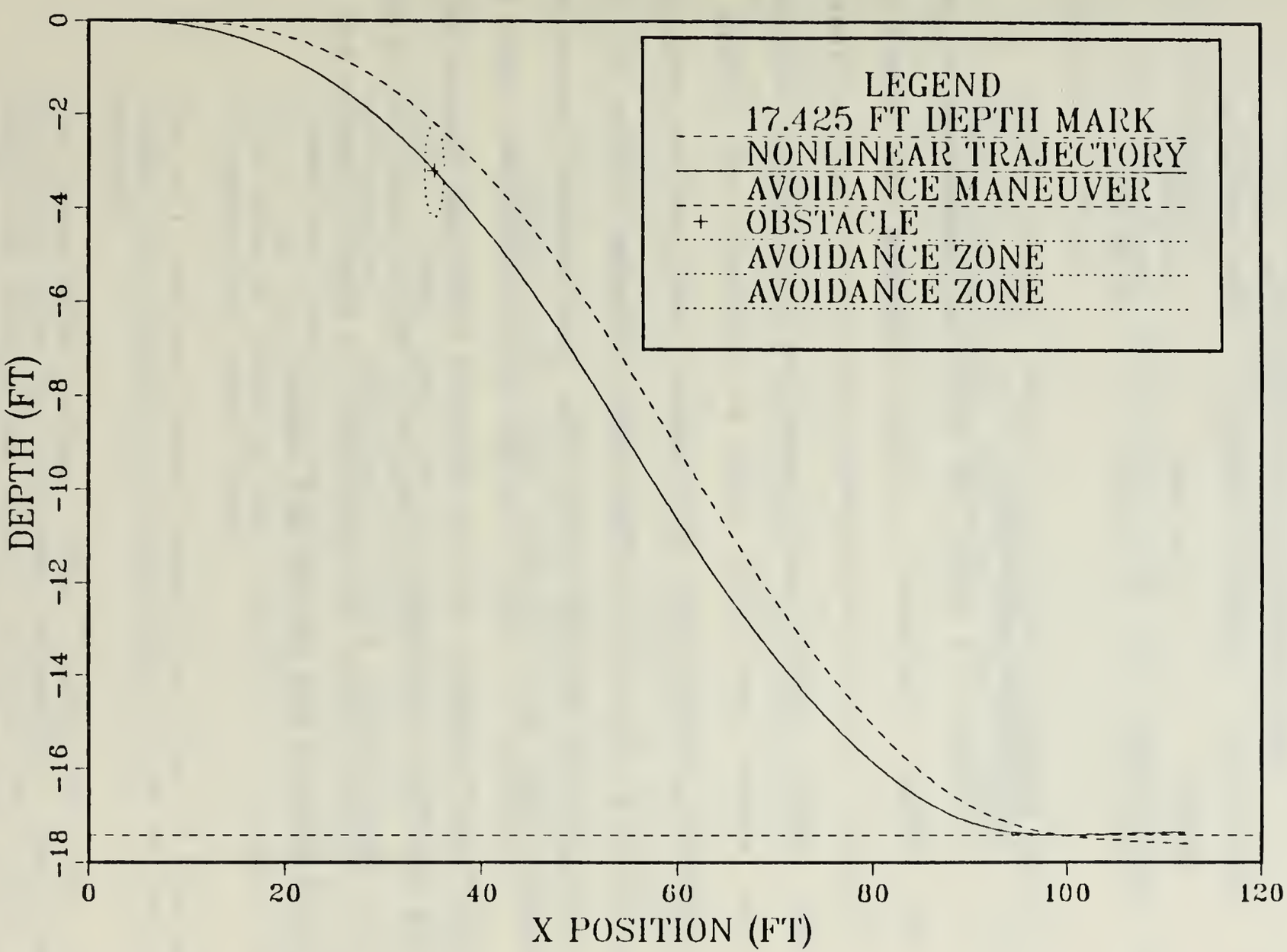


Figure 4.6 Avoidance Route with Avoidance Zone

TABLE 9

LINEAR/NONLINEAR OBSTACLE AVOIDANCE TIMES

MODEL TYPE	WITHOUT OBSTACLE	WITH OBSTACLE
LINEAR	3.12 SEC	3.50 SEC
NONLINEAR	22.57 SEC	48.67 SEC

2. Trajectory Analysis

Figure 4.3 shows the obstacle avoidance path above the optimal trajectory. This indicates that there are limits to the vehicle trajectory due to the vehicle dynamics and motion restrictions placed on the control surfaces. This effect was found when an obstacle was placed at different points along the optimal path and the avoidance trajectory was always above the obstacle. This is true since the ADS end constraints that are "driving" the vehicle are attempting to satisfy both of the weighted depth and pitch constraints. With an obstacle along the optimum path, the vehicles' maneuverability characteristics come into play --it becomes harder to take pitch off than it was to put it on to achieve the desired end state.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The following conclusions can be drawn from the feasibility study:

1. Optimal control theory is a feasible method of path planning with or without fixed obstacles. This study demonstrated successful path planning using the following:

   a) the objective function

$$\int_0^T \{[\underset{\sim}{x}_f(t)-\underset{\sim}{x}(t)]^T \underset{\sim}{Q}[\underset{\sim}{x}_f(t)-\underset{\sim}{x}(t)] + \underset{\sim}{u}(t)^T \underset{\sim\sim}{R}\underset{\sim}{u}(t)\}dt \ ,$$

   b) ADS equality constraints on desired end states,

   c) ADS inequality constraints for obstacle avoidance,

   d) a fixed step integration method (Rectangular) to integrate the rate equations and the objective function (time step = 0.1 sec, 10 intervals).

2. Real time path planning may be achieved with linear models. "Real time" here is arbitrarily determined to be from 1 to 2 minutes for the full 12 state optimization. Additional timing studies need to be done in order to determine this more precisely. The linearized dive plane equations of motion were studied and it was determined that a solution for a 20 second dive maneuver required three seconds of computer time to produce the "best" path.

3. The full nonlinear model computation time for the same linear dive maneuver was determined to be at most 22 seconds.

4. The simplified linear state trajectory was not compatible with the nonlinear trajectory. The linear model for dive plane motion exceeded the realm of small perturbation theory enough to be considered as a very poor substitute for planning the path of a real

vehicle in the dive plane under the assumption of decoupled dive plane motion.

5. The ADS optimization methods that gave the best results in the unconstrained cases was the Fletcher-Reeves method (014) and for the constrained cases was the modified method of feasible directions (057). Introduction of the strategy option provided less accuracy and more run time in the unconstrained case, and good accuracy and additional time in the constrained case.

6. The best method of achieving the trajectory between the start point and terminal point was to utilize the power of the ADS equality constraints to "drive" the trajectory to the desired final condition states. Weighting improved not only the accuracy but also run times.

7. The weighting values of the constraints in the linear model were not the "best" combination for the nonlinear model. Weighting was determined to be a function more of the plant model used than the size of the maneuvers the plant must undergo when using ADS constraints.

8. Obstacle avoidance was performed successfully using a simple algorithm that enforced constraints on the distance between the vehicle center of gravity and a given radius around the obstacle, called the "avoidance zone."

B. RECOMMENDATIONS

The following recommendations are offered to continue the development of a path planner algorithm for use in an autonomous underwater vehicle:

1. Carry out the procedures implemented here on such additional models as:

   a) the full nonlinear model to simulate other vehicle motion than the dive plane alone;

   b) the full linearized model in other motions,

   c) nonlinear reduced order models of the vehicle.

2. Conduct model sensitivity studies on the limits of maneuverability of the vehicle.

3. Continue development of the obstacle avoidance function to be able to have the model traverse a more dense environment of fixed and moving obstacles.

4. Develop the interface between the AUV command and control intelligence and the path planner, and between the vehicle controller and path planner.

5. Develop the program for eventual programming into a small microprocessor-based control structure.

APPENDIX A

EXAMPLE 10.2-6 FROM SAGE [REF. 19:P. 313]

Note: Solution presented here is based on the gradient in function space technique.

Consider the minimization of $J = \frac{1}{2}\int_0^1 (x^2 + u^2)dt$ for the system $\dot{x} = -x^2 + u$, $x(0) = 10.0$. We first need to determine the adjoint and the gradient equations. For this problem, the Hamiltonian is $H = \frac{1}{2}x^2 + \frac{1}{2}u^2 - \lambda x^2 + \lambda u$, and thus the adjoint equation is $\dot{\lambda} = -x + 2\lambda x$ with the terminal condition $\lambda(1) = 0$. The control gradient is $\partial H/\partial u = u + \lambda$. Suppose that we guess the initial control $u^o(t) = 0$, which is not too unreasonable, since the final value of the control, $u(1)$, should be zero, and use $K = 1$. To implement the gradient method, the steps we must take are:

1. Determine $x^N(t)$ from $u^N(t)$ for $t \in [0,1]$ by

$$\dot{x}^N = -[x^N(t)]^2 + u^N(t), \qquad x^N(0) = 10$$

2. Determine $\lambda^N(t)$ from $x^N(t)$ by

$$\dot{\lambda}^N = -x^N(t) + 2\lambda^N(t)x^N(t), \qquad \lambda^N(1) = 0$$

3. Determine $\partial H/\partial u^N$ from

$$\frac{\partial H}{\partial u^N} = u^N(t) + \lambda^N(t)$$

4. Determine $\Delta u^N(t)$ and ΔJ^N from

$$\Delta u^N(t) = -K \frac{\partial H}{\partial u^N} = - Ku^N(t) - K\lambda^N(t)$$

$$\Delta J^N(t) = -K \int_0^1 [\frac{\partial H}{\partial u^N}]^2 \, dt = -K \int_0^1 [u^N(t) + \lambda^N(t)]^2 \, dt$$

5. Compute the control for the next iteration

$$u^{N+1}(t) = u^N(t) + \Delta u^N(t)$$

6. Shuffle data and repeat the procedure, starting at 1, until $\Delta u^N(t)$ or $\Delta J^N(t)$ changes very little from iteration to iteration.

For the particular initial control assumed here, we obtain for the various steps in the procedure:

1. $u^o(t) = 0, \quad x^o(t) = 10/(1 + 10t)$
2. $\lambda^o(t) = \frac{1}{2}[1 - (1 + 10t)^2/121]$
3. $\partial H/\partial u^o = \frac{1}{2}[1 - (1 + 10t)^2/121]$
4. $\Delta u^o(t) = - \frac{1}{2}[1 - (1 + 10t)^2/121]$

$$\Delta J^o(t) = - \frac{1}{4} \int_0^1 [1 - (1 + 10t)^2/121]^2 dt$$

$$= - (0.0458)$$

5. $u^1(t) = u^o(t) + \Delta u^o(t) = -\frac{1}{2}[1 + 10t)^2/121]$.

Figures 2.1 and 2.2 are the curves generated by the equations above.

APPENDIX B

PROGRAMS

This appendix contains the three primary programs thatwere used for this feasibility study. They are:

1) STSPACE3 DSL--This is the state formulated DSL simulation of the linearized model for motion in the vertical plane.

2) THE1ND DSL--This is the ADSL program used to optimize the linear plant trajectories.

3) OPTT3 DSL--This program is the ADSL program used to optimize the full scale nonlinear model with the stern and bow planes decoupled.

PROGRAM NAME: STSPACE3 DSL

```
TITLE AUV VERTICAL PLANE MANEUVERS (STATE SPACE) REVISED
*
*       EQUILIBRIUM CONDITION IS CONSTANT SPEED (FT/SEC) IN
*       THE HORIZONTAL DIRECTION
*
CONST   UO=6.0
*
CONST   MA=         , THETAO=       , WO=           , IY=
*
CONST   ZW=-        , ZQ=           , ZQDOT=        , ZWDOT=
CONST   ZDB=        , ZDS=          , ZO=
*
CONST   MW=         , MQ=           , MQDOT=        , MWDOT=
CONST   MDB=        , MDS=          , MTHETA=
*
METHOD RECT
CONTROL FINTIM = 10., DELT= .1
INITIAL
      A1=-ZW
      A2=(MA-ZWDOT)
      A3=-(ZQ+MA)
      A4=-ZQDOT
      B1=-MW
      B2=-MWDOT
      B3=-MQ
      B4=(IY-MQDOT)
      B5=-MTHETA
      C1=ZDS
```

```
      C2=ZDB
      C3=MDS
      C4=MDB
      C5=C3-(B4*C1)/A4
      C6=C4-(B4*C2)/A4
      D1=B2-(B4*A2)/A4
      D2=B1-(B4*A1)/A4
      D3=B3-(B4*A3)/A4
*
DERIVATIVE
*
      THETDD=(1/A4)*(C1*DS+C2*DB-A1*W-A2*WDOT-A3*THETAD)
      WDOT=  (1/D1)*(C5*DS  +  C6*DB-  D2*W  -  D3*THETAD  -
      B5*THETA)
      THETAD= INTGRL(THETAO,THETDD)
      THETA = INTGRL(THETAO,THETAD)
      W = INTGRL(WO,WDOT)
      Z = INTGRL(ZO,W-(UO/UO)*THETA)
      DEPTH=-Z
      PITCH=THETA/.01745
      BOWANG=(DB/.01745)
      STNANG=(DS/.01745)
*
DYNAMIC
*
      DS  =  .08725*STEP(0.0)-.02843*STEP(2.0)-...
           .003636*STEP(3.0)-.008036*STEP(4.0)-...
           .015477*STEP(5.0)-.02670*STEP(6.0)-...
           .04007*STEP(7.0)-.053332*STEP(8.0)-
           .086084*STEP(9.0)

      DB  =  -.2443*STEP(0.0)  +  .3186*STEP(8.0)  +
           .17*STEP(9.0)
*
TERMINAL
*
PRINT  1.,THETDD,THETAD,THETA,ZDD,W,Z,DEPTH,PITCH,DS...
         ,BOWANG,STNANG
SAVE  .5,THETDD,THETAD,THETA,ZDD,W,Z,DEPTH,PITCH,DS...
         ,BOWANG,STNANG
GRAPH(DE=TEK618) TIME,DS
GRAPH(DE=TEK618) TIME,DEPTH
GRAPH(DE=TEK618) TIME,WDOT
GRAPH(DE=TEK618) TIME,W
GRAPH(DE=TEK618) TIME,THETDD
GRAPH(DE=TEK618) TIME,THETAD
GRAPH(DE=TEK618) TIME,THETA
GRAPH(DE=TEK618) TIME,PITCH
GRAPH(DE=TEK618) TIME,BOWANG
GRAPH(DE=TEK618) TIME,STNANG
END
STOP
```

NOTE: Coefficients must be supplied for the particular vehicle being studied.

```
PROGRAM NAME:   THE1ND DSL

TITLE RUN:(NR)  LINEAR AUV DYNAMIC PATH PLANNER FOR VERTICAL
      PLANE MOTION
* SEPARATED BOW AND STERN PLANE CONTROL NON-DIMENSIONAL
* DATA FOR NEW OBJECTIVE THAT INCLUDES THE ERROR UNWEIGHTED
* UPDATE:                    **OBSTACLE AVOIDANCE**
* USING OBJ= INTGRL?(Z-10)**2+W**2+THETA**2+THETAD**2+U**2
* OBJECTIVE FUNCTION WITH ADS CONSTRAINTS
********************** ADSL SET UP******************
*
FIXED ISTRAT, IOPT, IONED, IPRINT, INFO, IGRAD, NDV, NCON
FIXED IDG, NGT, IC, NRA, NCOLA, NRWK, IWK, NRIWK, O, H
D     DIMENSION AW(42,42)
ARRAY WK(5000), IWK(1000),DIST(15)
ARRAY DX(21),VLB(21),VUB(21),GW(15), DF(21), IDG(15), IC(20)
PARAM NRA=42, NCOLA=42, NRWK=5000, NRIWK=1000
PARAM IGRAD=0, INFO=0, NDV=20, NCON=15, NGT=20
TABLE DX(1-2)=2*.0, DX(3-21)=19*0., IDG(1-4)=4*-1
TABLE IDG(5-15)=11*0
TABLE VLB(1-9)=10*-.17452, VLB(11-19)=9*-.2443
TABLE VLB(10)=0.,VLB(20-21)=0.
TABLE VUB(1-9)=9*.17452, VUB(11-19)=9*.2443
TABLE VUB(10)=0.,VUB(20-21)=0.
TABLE DIST(15)=15*0.0
PARAM ISTRAT=0, IOPT=5, IONED=7, IPRINT=2020
INCON U=0., H=0
METHOD RECT
CONTROL FINTIM= 7.00, DELT=.1
PRINT THETAD,W,DEPTH,PITCH,XPOS,ZPOS,DT
*
*****************************DSL MODEL SET UP***************
*
*       EQUILIBRIUM CONDITION IS CONSTANT SPEED (NON-
*       DIMENSIONALIZED) BY  UO = 6 FT/SEC
*
CONST UO=1.0, XOBS=       ,ZOBS=
*
CONST  MA=0.0962, THETAO=0., ZO=0.,WO=0., IY=.00606
*
CONST  ZW=      , ZQ=         , ZQDOT=         , ZWDOT=
CONST ZDB=      , ZDS=        , ZO=0.
*
CONST  MW=       , MQ=        , MQDOT=       , MWDOT=
CONST  MDB=      , MDS=       , MTHETA=
*
*
INITIAL
*
      ORDDEP = 1.00
      A1=-ZW
      A2=(MA-ZWDOT)
```

```
      A3=-(ZQ+MA)
      A4=-ZQDOT
      B1=-MW
      B2=-MWDOT
      B3=-MQ
      B4=(IY-MQDOT)
      B5=-MTHETA
      C1=ZDS
      C2=ZDB
      C3=MDS
      C4=MDB
      C5=C3-(B4*C1)/A4
      C6=C4-(B4*C2)/A4
      D1=B2-(B4*A2)/A4
      D2=B1-(B4*A1)/A4
      D3=B3-(B4*A3)/A4
      CALL DADS(INFO,ISTRAT,IOPT,IONED,IPRINT,IGRAD,...
           NDV,NCON,DX,VLB,VUB,OBJ,GW,IDG,NGT,IC,DF,...
           AW,NRA,NCOLA,WK,NRWK,IWK,NRIWK)
      IF(INFO.EQ.0) DELPRT = 0.2
      IF(INFO.EQ.0) DELPLT = 0.2
*
DERIVATIVE
*
      THETDD=(1/A4)*(C1*DS+C2*DB-A1*W-A2*WDOT-A3*THETAD)
      WDOT= (1/D1)*(C5*DS + C6*DB - D2*W - D3*THETAD - ...
             B5*THETA)
      THETAD= INTGRL(THETAO,THETDD)
      THETA = INTGRL(THETAO,THETAD)
      W = INTGRL(WO,WDOT)
      Z = INTGRL(ZO,W-UO*THETA)
      DEPTH=-Z
      PITCH=THETA/.01745
      BOWANG=(DB/.01745)
      STNANG=(DS/.01745)
      INTGRD = ((W*W+(Z-ORDDEP)*(Z-ORDDEP)+...
               THETAD*THETAD+THETA*THETA)) + (DS*DS+DB*DB)
      OBJ1 = INTGRL(0.,(0.5)*INTGRD)
      OBJ = OBJ1
*
DYNAMIC
*
      RN=TIME/(FINTIM/10.-DELT/10000.)
      O=INT(RN)+1
      IF(O.EQ.11) O=10
*
* ADDITIONALLY THE PLANES SHOULD BE AT EQUILIBRIUM SO THE
* VEHICLE WILL PROCEED AT THIS NEW DEPTH WITHIN SOME
* TOLERANCE
*
      DS=DX(O)
      DB=DX(10+O)
```

```
      IF(O.GE.10) DS=0.
      IF(O.GE.10) DB=0.
*
*         CONSTRAINTS FOR A DIVE
*
*  ORDERED DEPTH = ORDDEP
      GW(1) =(Z-ORDDEP)/2.
      GW(2) =(ORDDEP-Z)/2.
*
*  AUV'S FINAL STATE MUST BE LEVEL FLIGHT AS FOLLOWS
*
      GW(3) = THETA*10.
      GW(4) = -THETA*10.
*
*   X-Z POSITIONING FOR OBSTACLE AVOIDANCE
      XPOS=17.425*TIME
      ZPOS=-Z*17.425
*
*               AVOIDING     THE     OBSTACLE
      IF(TIME.GE.0.0.AND.TIME.LE.XOBS/17.425) THEN
      TIME1 = XOBS/17.425
      QN=TIME/(TIME1/10.-DELT/10000.)
      D = I N T ( Q N ) + 1
      DIST(D)=SQRT((XPOS-XOBS)*(XPOS-XOBS)+(ZPOS-ZOBS)*...
              (ZPOS-ZOBS))
      GW(4+D)  = (1.-DIST(D))
      ELSE
      ENDIF
*
TERMINAL
      IF(INFO.EQ.0) THEN
      DO 9000 D=1,11
      WRITE(*,9999) DIST(D)
9999 FORMAT(1X,E15.4)
9000 CONTINUE
     ELSE
     ENDIF
     IF(INFO.EQ.0) CALL ENDJOB
     CALL RERUN
*
END
STOP
```

NOTE: The coefficients must be provided for the particular vehicle being studied.

```
PROGRAM NAME:   OPTT3 DSL

TITLE RUN: NONLINEAR AUV MODEL / STERN PLANE AND BOW PLANE
          SEPARATED
* (1) UPDATED:11/16/87
* (2) 100.00 FT DEPTH CHANGE IN 20 SEC
* (3) RIGHT OBJ EQUATION
* (4) ADS CONSTRAINTS ON DEPTH AND PITCH
* (5) OBSTACLE FURTHER DOWN THE TRAJECTORY AND ABOVE IT
*
**********************ADSL SET-UP*****************************
*FIXED ISTRAT, IOPT, IONED, IPRINT, INFO, IGRAD, NDV, NCON
FIXED IDG, NGT, IC, NRA, NCOLA, NRWK, IWK, NRIWK, O, H,D,C
D DIMENSION AW(82,82)
ARRAY WK(5000), IWK(1000)
ARRAY DIST(40)
ARRAY DX(21),VLB(21),VUB(21),GW(15), DF(21), IDG(15), IC(20)
PARAM NRA=82, NCOLA=82, NRWK=5000, NRIWK=1000
PARAM IGRAD=0, INFO=0, NDV=20, NCON=15, NGT=20
TABLE DX(1-2)=2*.0,DX(3-21)=19*0., IDG(1-4)=4*-1
TABLE IDG(5-15)=11*0
TABLE VLB(1-9)=9*-.17452, VLB(11-19)=9*-.2443
TABLE VLB(10)=0.,VLB(20-21)=0.
TABLE VUB(1-9)=9*.17452, VUB(11-19)=9*.2443
TABLE VUB(10)=0.,VUB(20-21)=0.TABLE DIST(1-40)=40*0.
PARAM ISTRAT=0, IOPT=5, IONED=7, IPRINT=2020
INCON  H=0, OBS1=0.,YZONE=0.
METHOD RECT
CONTROL FINTIM=20., DELT=.10
PRINT THETAD,W,DEPTH,PITCH,XPOS,DEPTH,NDX,NDZ,NDT
*
*********************DSL MODEL SET UP************************
*
D DIMENSION MM(6,6),G4(4),GK4(4),BR(4),HH(4)
D       DIMENSION  B(6,6),BB(6,6)
D       DIMENSION A(12,12), AA(12,12)
D       COMMON / BLOCK1 / F(12), FP(6), MMINV(6,6), UCF(4)
FIXED  N,IA,IDGT,IER,LAST,J,K,M,JJ,KK,I
INTEGER
ARRAY  WKAREA(54), X(12)
*
*
CONST
*
*      LONGITUDINAL HYDRODYNAMIC COEFFICIENTS
*
CONST XPP =    ,XQQ =        ,XRR =        ,XPR =              ...
      XUDOT=   ,XWQ =        ,XVP =        ,XVR =              ...
      XQDS=    ,XQDB=        ,XRDR=        ,XVV =              ...
      XWW =    ,XVDR=        ,XWDS=        ,XWDB=              ...
      XDSDS=   ,XDBDB=      ,XDRDR=      ,XQDSN=              ...
      XWDSN=   ,XDSDSN=
```

```
*
*      LATERAL HYDRODYNAMIC COEFFICIENTS
*
CONST YPDOT=   ,YRDOT=      ,YPQ =       ,YQR =              ...
      YVDOT=   ,YP  =       ,YR =        ,YVQ =              ...
      YWP =    ,YWR =       ,YV =        ,YVW =              ...
      YDR =    ,CDY =
*
*      NORMAL HYDRODYNAMIC COEFFICIENTS
*
CONST ZQDOT=   ,ZPP =      ,ZPR =          ,ZRR =           ...
      ZWDOT=   ,ZQ =       ,ZVP =          ,ZVR =           ...
      ZW =     ,ZVV =      ,ZDS =          ,ZDB =           ...
      ZQN =    ,ZWN =     ,ZDSN=          ,CDZ =
*
*      ROLL HYDRODYNAMIC COEFFITIENTS
*
CONST KPDOT=   ,KRDOT=     ,KPQ =          ,KQR =           ...
      KVDOT=   ,KP =       ,KR =           ,KVQ=            ...
      KWP =    ,KWR =      ,KV =           ,KVW =           ...
      KPN =    ,KDB =
*
*      PITCH HYDRODYNAMIC COEFFICIENTS
*
CONST MQDOT=   ,MPP =      ,MPR =          ,MRR =           ...
      MWDOT=   ,MQ =       ,MVP =          ,MVR =           ...
      MW =     ,MVV =      ,MDS =          ,MDB =           ...
      MQN =    ,MWN =     ,MDSN =
*
*      YAW HYDRODYNAMIC COEFFICIENTS
*
CONST NPDOT=   ,NRDOT=     ,NPQ =          ,NQR =           ...
      NVDOT=   ,NP =       ,NR =           ,NVQ =           ...
      NWP =    ,NWR =      ,NV =           ,NVW =           ...
      NDR =
*
*      MASS CHARACTERISTICS OF THE FLOODED MARK IX VEHICLE
*
CONST WEIGHT =   , BOY =       ,VOL =         ,XG =         ...
      YG =       , ZG =        ,XB =          ,ZB =         ...
      IX =       , IY =        ,IZ =          ,IXZ =        ...
      IYZ =      , IXY =       ,YB =                        ...
      L =        , RHO =       ,G =           ,NU =         ...
      A0 =       ,KPROP =      ,NPROP =     ,X1TEST=        ...
      DEGRUD= ,DEGSTN=
*
*      INPUT INITIAL CONDITIONS HERE IF REQUIRED
*
INITIAL
NOSORT
      D=0
      H=H+1
```

```
      ORDDEP = 17.425
      XOBS=58.666
      ZOBS=6.0000
      IF(H.EQ.1) THEN
      U = 0.0
      V = 0.0
      W = 0.0
      P = 0.0
      Q = 0.0
      R = 0.0
      XPOS = 0.0
      YPOS = 0.0
      ZPOS = 0.0
      PSI = 0.0
      THETA = 0.0
      PHI = 0.0
*
      U0 = 6.0
      V0 = 0.0
      W0 = 0.0
      P0 = 0.0
      Q0 = 0.0
      R0 = 0.0
      PHI0 = 0.0
      THETA0 = 0.0
      PSI0 = 0.0
      DB= 0.0
      DS = 0.0
      DR = 0.0
      RPM = 500
      LATYAW = 0.0
      NORPIT = 0.0
      RE = U0*L/NU
      CD0 =  .00385 + (1.296E-17)*(RE - 1.2E7)**2
*
*     DEFINE LENGTH FRACTIONS FOR GAUSS QUADUTURE TERMS
*
      G4(1) = 0.069431844
      G4(2) = 0.330009478
      G4(3) = 0.669990521
      G4(4) = 0.930568155
*
*     DEFINE WEIGHT FRACTIONS FOR GAUSS QUADUTURE TERMS
*
      GK4(1) = 0.1739274225687
      GK4(2) = 0.3260725774312
      GK4(3) = 0.3260725774312
      GK4(4) = 0.1739274225687
*
*       DEFINE THE BREADTH BB AND HEIGHT HH TERMS FOR THE
      INTEGRATION
*
```

```
      BR(1) = 75.7/12
      BR(2) = 75.7/12
      BR(3) = 75.7/12
      BR(4) = 55.08/12
*
      HH(1) = 16.38/12
      HH(2) = 31.85/12
      HH(3) = 31.85/12
      HH(4) = 23.76/12
*
      MASS = WEIGHT/G
*
      DIVAMP = DEGSTN*0.0174532925
      RUDAMP = DEGRUD*0.0174532925
*
*
      N = 6
      DO 15 J = 1,N
          DO 10 K = 1,N
          MMINV(J,K) = 0.0
          MM(J,K) = 0.0
10        CONTINUE
15    CONTINUE
*
*
*
*
*
*
      MM(1,1) = MASS -((RHO/2)*(L**3)*XUDOT)
      MM(1,5) = MASS*ZG
      MM(1,6) = -MASS*YG
*
      MM(2,2) = MASS -((RHO/2)*(L**3)*YVDOT)
      MM(2,4) = -MASS*ZG -((RHO/2)*(L**4)*YPDOT)
      MM(2,6) = MASS*XG - ((RHO/2)*(L**4)*YRDOT)
*
      MM(3,3) = MASS - ((RHO/2)*(L**3)*ZWDOT)
      MM(3,4) = MASS*YG
      MM(3,5) = -MASS*XG -((RHO/2)*(L**4)*ZQDOT)
*
      MM(4,2) = -MASS*ZG - ((RHO/2)*(L**4)*KVDOT)
      MM(4,3) = MASS*YG
      MM(4,4) = IX - ((RHO/2)*(L**5)*KPDOT)
      MM(4,5) = -IXY
      MM(4,6) = -IXZ -((RHO/2)*(L**5)*KRDOT)

      MM(5,1) = MASS*ZG
      MM(5,3) = -MASS*XG -((RHO/2)*(L**4)*MWDOT)
      MM(5,4) = -IXY
      MM(5,5) = IY -((RHO/2)*(L**5)*MQDOT)
      MM(5,6) = -IYZ
```

```
*
      MM(6,1) = -MASS*YG
      MM(6,2) = MASS*XG -((RHO/2)*(L**4)*NVDOT)
      MM(6,4) = -IXZ - ((RHO/2)*(L**5)*NPDOT)
      MM(6,5) = -IYZ
      MM(6,6) = IZ - ((RHO/2)*(L**5)*NRDOT)
*
*
      LAST = N*N+3*N
      DO 20 M = 1,LAST
      WKAREA(M) = 0.0
20    CONTINUE
*
      IER = 0
      IA = 6
      IDGT = 4
      WRITE( 8,400)((MM(I,J), J = 1,6),I = 1,6)
*
      CALL LINV2F(MM,N,IA,MMINV,IDGT,WKAREA,IER)
*
      WRITE( 8,400)((MMINV(I,J), J = 1,6),I = 1,6)
400   FORMAT(6E12.4)
*
      ELSE
      ENDIF
*
*
      CALL  DADS(INFO,ISTRAT,IOPT,IONED,IPRINT,IGRAD,...
      NDV,NCON,DX,VLB,VUB,OBJ,GW,IDG,NGT,IC,DF,AW,NRA,...
      NCOLA,WK,NRWK,IWK,NRIWK)
      IF(INFO.EQ.0) DELPRT = 1.0
*
DERIVATIVE
NOSORT
*
*     PROPULSION MODEL
*
      SIGNU = 1.0
      IF (U.LT.0.0) SIGNU = -1.0
      IF (ABS(U).LT.X1TEST) U = X1TEST
      SIGNN = 1.0
      IF (RPM.LT.0.0) SIGNN = -1.0
      ETA = 0.012*RPM/U
      RE = U*L/NU
      CD0 =  .00385 + (1.296E-17)*(RE - 1.2E7)**2
      CT  = 0.008*L**2*ETA*ABS(ETA)/(A0)
      CT1 = 0.008*L**2/(A0)
      EPS = -1.0+SIGNN/SIGNU*(SQRT(CT+1.0)-...
                  1.0)/(SQRT(CT1+1.0)-1.0)
      XPROP = CD0*(ETA*ABS(ETA) -1.0)
*
*     CALCULATE THE DRAG FORCE, INTEGRATE THE DRAG OVER THE
```

```
*     VEHICLE AND INTEGRATE USING A 4 TERM GAUSS QUADUTURE
*
      LATYAW = 0.0
      NORPIT = 0.0
      DO 500 K = 1,4
          UCF(K) = SQRT((V+G4(K)*R*L)**2 + (W-G4(K)*Q*L)**2)
          IF(UCF(K).GT.1E-10) THEN
          TERM0  = (RHO/2)*(CDY*HH(K)*(V+G4(K)*R*L)**2 +...
                    CDZ*BR(K)*(W-G4(K)*Q*L)**2)
          TERM1  = TERM0*(V+G4(K)*R*L)/UCF(K)
          TERM2  = TERM0*(W-G4(K)*Q*L)/UCF(K)
          LATYAW = LATYAW + TERM1*GK4(K)*L
          NORPIT = NORPIT + TERM2*GK4(K)*L
          END IF
500   CONTINUE
*
*
*     FORCE EQUATIONS
*
*
*     LONGITUDINAL FORCE
*
      FP(1) =MASS*V*R-MASS*W*Q+MASS*XG*Q**2+MASS*XG*R**2-...
             MASS*YG*P*Q-MASS*ZG*P*R+(RHO/2)*L**4*          ...
             (XPP*P**2 + XQQ*Q**2 + XRR*R**2 + XPR*P*R)     ...
             +(RHO/2)*L**3*(XWQ*W*Q +XVP*V*P+XVR*V*R+U*Q    ...
             *(XQDS*DS+XQDB*DB)+XRDR*U*R* DR)+              ...
             (RHO/2)*L**2*(XVV*V**2 + XWW*W**2 +            ...
             XVDR*U*V*DR + U*W*(XWDS*DS+XWDB*DB)+U**2*      ...
             (XDSDS*DS**2+XDBDB*DB**2 + XDRDR*DR**2))-      ...
             (WEIGHT -BOY)*SIN(THETA) +(RHO/2)*L**3*        ...
             XQDSN*U*Q*DS*EPS+(RHO/2)*L**2*(XWDSN*U*W*DS+...
             XDSDSN*U**2*DS**2)*EPS +(RHO/2)*L**2*U**2*XPROP
*
*     LATERAL FORCE
*
      FP(2) = -MASS*U*R + MASS*XG*P*Q + MASS*YG*R**2 -     ...
             MASS*ZG*Q*R +(RHO/2)*L**4*(YPQ*P*Q +          ...
             YQR*Q*R)+(RHO/2)*L**3*(YP*U*P + YR*U*R +      ...
             YVQ*V*Q + YWP*W*P + YWR*W*R) + (RHO/2)*L**2*...
             (YV*U*V + YVW*V*W +YDR*U**2*DR) -LATYAW       ...
             +(WEIGHT-BOY)*...  COS(THETA)*SIN(PHI)
*
*     NORMAL FORCE
*
      FP(3) = MASS*U*Q - MASS*V*P - MASS*XG*P*R -           ...
             MASS*YG*Q*R +MASS*ZG*P**2 + MASS*ZG*Q**2 + ...
             (RHO/2)*L**4*(ZPP*P**2 + ZPR*P*R +ZRR*R**2)...
             + (RHO/2)*L**3*(ZQ*U*Q + ZVP*V*P + ZVR*V*R)...
             +(RHO/2)*L**2*(ZW*U*W + ZVV*V**2+U**2*(ZDS*...
             DS+ZDB*DB))-NORPIT+(WEIGHT-BOY)*COS(THETA)*...
             COS(PHI)+ (RHO/2)*L**3*ZQN*U*Q*EPS            ...
```

```
     +(RHO/2)*L**2*(ZWN*U*W +ZDSN*U**2*DS)*EPS
*
*    ROLL FORCE
*
      FP(4) =-IZ*Q*R+IY*Q*R-IXY*P*R +IYZ*Q**2 -IYZ*R**2 ...
             +IXZ*P*Q +MASS*YG*U*Q -MASS*YG*V*P -          ...
             MASS*ZG*W*P+(RHO/2)*L**5*(KPQ*P*Q+ KQR*Q*R)...
             +(RHO/2)*L**4*(KP*U*P +KR*U*R + KVQ*V*Q +    ...
             KWP*W*P + KWR*W*R) +(RHO/2)*L**3*(KV*U*V +   ...
             KVW*V*W) +(YG*WEIGHT -YB*BOY)*COS(THETA)*    ...
             COS(PHI) -(ZG*WEIGHT -ZB*BOY)*COS(THETA)*    ...
             SIN(PHI) + (RHO/2)*L**4*KPN*U*P*EPS+         ...
             (RHO/2)*L**3*U**2*KPROP +MASS*ZG*U*R
*
*    PITCH FORCE
*
      FP(5) = -IX*P*R +IZ*P*R+IXY*Q*R-IYZ*P*Q-IXZ*P**2    ...
             +IXZ*R**2 -MASS*XG*U*Q + MASS*XG*V*P +        ...
             MASS*ZG*V*R -MASS*ZG*W*Q +                    ...
             (RHO/2)*L**5*(MPP*P**2 +MPR*P*R               ...
             +MRR*R**2)+(RHO/2)*L**4*(MQ*U*Q + MVP*V*P +...
             MVR*V*R) + (RHO/2)*L**3*(MW*U*W +             ...
             MVV*V**2+U**2*(MDS*DS+MDB*DB))+ NORPIT -      ...
             (XG*WEIGHT- XB*BOY)*COS(THETA)*COS(PHI) +     ...
             (RHO/2)*L**3*(MWN*U*W+MDSN*U**2*DS)*EPS-      ...
             (ZG*WEIGHT-ZB*BOY)*SIN(THETA)
*
*    YAW FORCE
*
      FP(6) = -IY*P*Q+IX*P*Q+IXY*P**2-IXY*Q**2+IYZ*P*R -...
             IXZ*Q*R -MASS*XG*U*R + MASS*XG*W*P -          ...
             MASS*YG*V*R + MASS*YG*W*Q +                   ...
             (RHO/2)*L**5*(NPQ*P*Q + NQR*Q*R)              ...
             +(RHO/2)*L**4*(NP*U*P+NR*U*R + NVQ*V*Q        ...
             +NWP*W*P + NWR*W*R) +(RHO/2)*L**3*(NV*U*V +...
             NVW*V*W+NDR*U**2*DR)- LATYAW + (XG*WEIGHT -...
             XB*BOY)*COS(THETA)*SIN(PHI)+(YG*WEIGHT)*      ...
             SIN(THE TA)+(RHO/2)*L**3*U**2*NPROP-          ...
             YB*BOY*SIN(THETA)
*
*    NOW COMPUTE THE F(1-6) FUNCTIONS
*
      DO 600 J = 1,6
            F(J) = 0.0
      DO 600 K = 1,6
            F(J) = MMINV(J,K)*FP(K) + F(J)
600   CONTINUE
*
*    THE LAST SIX EQUATIONS COME FROM THE KINEMATIC
*              RELATIONS
*
*    FIRST SET THE DRIFT CURRENT VALUES
```

```
*
      UCO = 0.0
      VCO = 0.0
      WCO = 0.0
*
*     INERTIAL POSITION RATES F(7-9)
*
      F(7) = UCO + U*COS(PSI)*COS(THETA) +                    ...
             V*(COS(PSI)*SIN(THETA)*                          ...
             SIN(PHI) - SIN(PSI)*COS(PHI)) +                  ...
             W*(COS(PSI)*SIN(THETA)*COS(PHI)+                 ...
             SIN(PSI)*SIN(PHI))
*
      F(8) = VCO + U*SIN(PSI)*COS(THETA) +                    ...
             V*(SIN(PSI)*SIN(THETA)*SIN(PHI) +                ...
             COS(PSI)*COS(PHI))+ W*(SIN(PSI)*SIN(THETA)*...
             COS(PHI) -COS(PSI)*SIN(PHI))
*
      F(9) = WCO - U*SIN(THETA) +V*COS(THETA)*SIN(PHI)   ...
             +W*COS(THETA)*COS(PHI)
*
*     EULER ANGLE RATES F(10-12)
*
      F(10) = P + Q*SIN(PHI)*TAN(THETA)+                      ...
             R*COS(PHI)*TAN(THETA)
*
      F(11) = Q*COS(PHI) - R*SIN(PHI)
*
      F(12) = Q*SIN(PHI)/COS(THETA) + R*COS(PHI)/COS(THETA)

*
      UDOT = F(1)
      VDOT = F(2)
      WDOT = F(3)
      PDOT = F(4)
      QDOT = F(5)
      RDOT = F(6)
      XDOT = F(7)
      YDOT = F(8)
      ZDOT = F(9)
      PHIDOT = F(10)
      THETAD = F(11)
      PSIDOT = F(12)
*
      U = INTGRL (U0,UDOT)
      V = INTGRL(0.0,VDOT)
      W = INTGRL(0.0,WDOT)
      P = INTGRL(0.0,PDOT)
      Q = INTGRL(0.0,QDOT)
      R = INTGRL(0.0,RDOT)
      XPOS = INTGRL(0.0,XDOT)
      YPOS = INTGRL(0.0,YDOT)
```

```
      ZPOS = INTGRL(0.0,ZDOT)
      PHI = INTGRL(0.0,PHIDOT)
      THETA = INTGRL(0.0,THETAD)
      PSI = INTGRL(0.0,PSIDOT)
*
      PHIANG = PHI/0.0174532925
      THEANG = THETA/0.0174532925
      PSIANG = PSI/0.0174532925
*
      DEPTH=-ZPOS
      PITCH=THEANG
      BOWANG=(DB/.01745)
      STNANG=(DS/.01745)
      INTGRD = (U*U+V*V+W*W+P*P+Q*Q+R*R+XPOS*XPOS+YPOS*...
               YPOS+(ZPOS-ORDDEP)*(ZPOS-ORDDEP)+PHI*PHI+...
               THETA*THETA+PSI*PSI) + (DS*DS+DB*DB)
      OBJ1 = INTGRL(0.,(0.5)*INTGRD)
      OBJ = OBJ1
*
DYNAMIC
      RN=TIME/(FINTIM/10.-DELT/10000.)
      O=INT(RN)+1
      IF(O.EQ.11) O=10
*
* ADDITIONALLY THE PLANES SHOULD BE AT EQUILIBRIUM SO THE
* VEHICLE WILL PROCEED AT THIS NEW DEPTH WITHIN SOME
*      TOLERANCE
*
      DS=DX(O)
      DB=DX(10+O)
      IF(O.GE.10) DS=0.
      IF(O.GE.10) DB=0.
*     DR=DX(20+O)
*     RPM=DX(30+O)
*
*         CONSTRAINTS FOR A DIVE
*
*  ORDERED DEPTH = ORDDEP
      GW(1) = (ZPOS-ORDDEP)/5.
      GW(2) = (ORDDEP-ZPOS)/5.
*  AUV'S FINAL STATE MUST BE LEVEL FLIGHT AS FOLLOWS
      GW(3) = THETA/5.
      GW(4) = -THETA/5.
*
*   AVOIDING THE OBSTACLE
*
      IF(TIME.GE.0.0.AND.TIME.LE.XOBS/U) THEN
      TIME1 = XOBS/U
      QN=TIME/(TIME1/10.-DELT/10000.)
      D = I N T ( Q N ) + 1
      DIST(D)=SQRT((XPOS-XOBS)*(XPOS-XOBS)+(ZPOS-
      ZOBS)*(ZPOS-ZOBS)) GW(4+D)  = (1.-DIST(D))
```

```
      ELSE
      ENDIF
      NDX=XPOS/17.425
      NDZ=-ZPOS/17.425
      NDT=TIME*6./17.425
*
TERMINAL
*
      IF(INFO.EQ.0) THEN
      DO 9000 D=1,11
      WRITE(*,9999) DIST(D)
9999 FORMAT(1X,E15.4)
9000 CONTINUE
     ELSE
     ENDIF
     IF(INFO.EQ.0) CALL ENDJOB
     CALL RERUN
*
END
STOP
```

NOTE: The coefficients must be supplied for the particular vehicle being studied.

LIST OF REFERENCES

1. Nitao, J.J. and Parodi, A.M., "A Real Time Reflexive Pilot for an Autonomous Land Vehicle," IEEE Control Systems Magazine, pp. 13-23, February 1986.

2. Crowley, J.L., "Navigation for an Intelligent Mobile Robot," IEEE Journal of Robitics and Automation, V. RA-1, No. 1, pp. 31-41, March 1985.

3. Chattergy, R., "Some Heuristics for the Navigation of a Robot," The International Journal of Robotics Research, V. 4, No. 1, Spring 1985.

4. Kambhampati, S. and Davis, L.S., "Multiresolution Path Planning for Mobile Robots," Computer Vision Laboratory Report, University of Maryland, College Park, May 1985.

5. Wong, E.K. and Fuk, K.S., "Hierarchical Orthogonal Space Approach to Three-Dimensional Path Planning," IEEE Journal of Robotics and Automation, V. RA-2, No. 1, March 1986.

6. Herman, M., "Fast, Three Dimensional, Collision-Free Motion Planning," Proceedings 1986 IEEE International Conference on Robotics and Automation, V. 2, pp. 1056-1063, April 1986.

7. Singh, S. and Wagh, M.D., "Robot Path Planning Using Intersecting Convex Shapes," Proceedings 1986 IEEE International Conference on Robotics and Automation, V. 3, pp. 1743-1748, April 1986.

8. Kuan, D.T., Zamiska, J.C. and Brooks, R.A., "Natural Decomposition of Free Space for Path Planning," Proceedings IEEE International Conference on Robotics and Automation, pp. 168-173, 1985.

9. Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," International Journal of Robotics Research, V. 5, No. 1, Spring 1986.

10. Tournassoud, P., "A Strategy for Obstacle Avoidance and Its Application to Multi-Robot Systems," Proceedings IEEE 1986 Conference on Robotics and Automation, V. 2, pp. 1224-1229, April 1986.

11. Krogh, B.H., "Guaranteed Steering Control," Proceedings of the 1985 American Control Conference, V. 2, pp. 950-955, June 1985.

12. Thorpe, C.E., "Path Relaxation: Path Planning for a Mobile Robot," CMU Robotics Institute Report CMU-RI-TR-84-5, April 1984.

13. Krogh, B.H. and Thorpe, C.E., "Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles," Proceedings 1986 International Conference on Robotics and Automation, V. 3, pp. 1664-1669, April 1986.

14. Gilbert, E.G. and Johnson, D.W., "Distance Functions and Their Application to Robot Path Planning in the Presence of Obstacles," IEEE Journal of Robotics and Automation, V. RA-1, No. 1, March 1985.

15. Johnson, D.W. and Gilbert, E.G., "Minimum Time Robot Path Planning in the Presence of Obstacles," Proceedings of the 24th Conference on Decision and Control, V. 3, pp. 1748-1753, December 1985.

16. Vanderplaats, G.N. and Sugimoto, H., ADS Design Optimazation Program, FORTRAN Source Code, Naval Postgraduate School, Monterey, California, June 1982.

17. International Business Machines Program Number 5798-PXJ, Dynamic Simulation Language, June 1984.

18. Boncal, R., A Study of Model Based Maneuvering Controls for Autonomous Underwater Vehicles, Master's Thesis, Naval Postgraduate School, Monterey, California, December 1987.

19. Sage, A.P. and White, C., Optimum Systems Control, Prentice-Hall, 1977.

20. Vanderplaats, G.N., ADS A Fortran Program for Automatic Design Synthesis, Version 1.10, Engineering Design Optimization, Inc., 1985.

21. Olson, T.F., Application of Numerical Optimization in Modern Control, Master's Thesis, Naval Postgraduate School, Monterey, California, December 1986.

22. Vanderplaats, G.N., Numerical Optimization Techniques for Engineering Design, McGraw-Hill, 1984.

23. Comstock, J.P., ed., Principles of Naval Architect, The Society of Naval Architects and Marine Engineers, New York, 1967.

24. Gertler, M. and Hagen, G.R., "Standard Equations for Submarine Simulation," Naval Ship Research and Development Center Report 2510, June 1967.

25. Feldman, J., "DTNSRDC Revised Standard Submarine Equations of Motion," Report Nr. DTNSRDC/SPD-0393-09, David W. Taylor Naval Ship Research and Development Center, June 1979.

26. Ogata, K., Modern Control Engineering, Prentice-Hall, 1970.

INITIAL DISTRIBUTION LIST

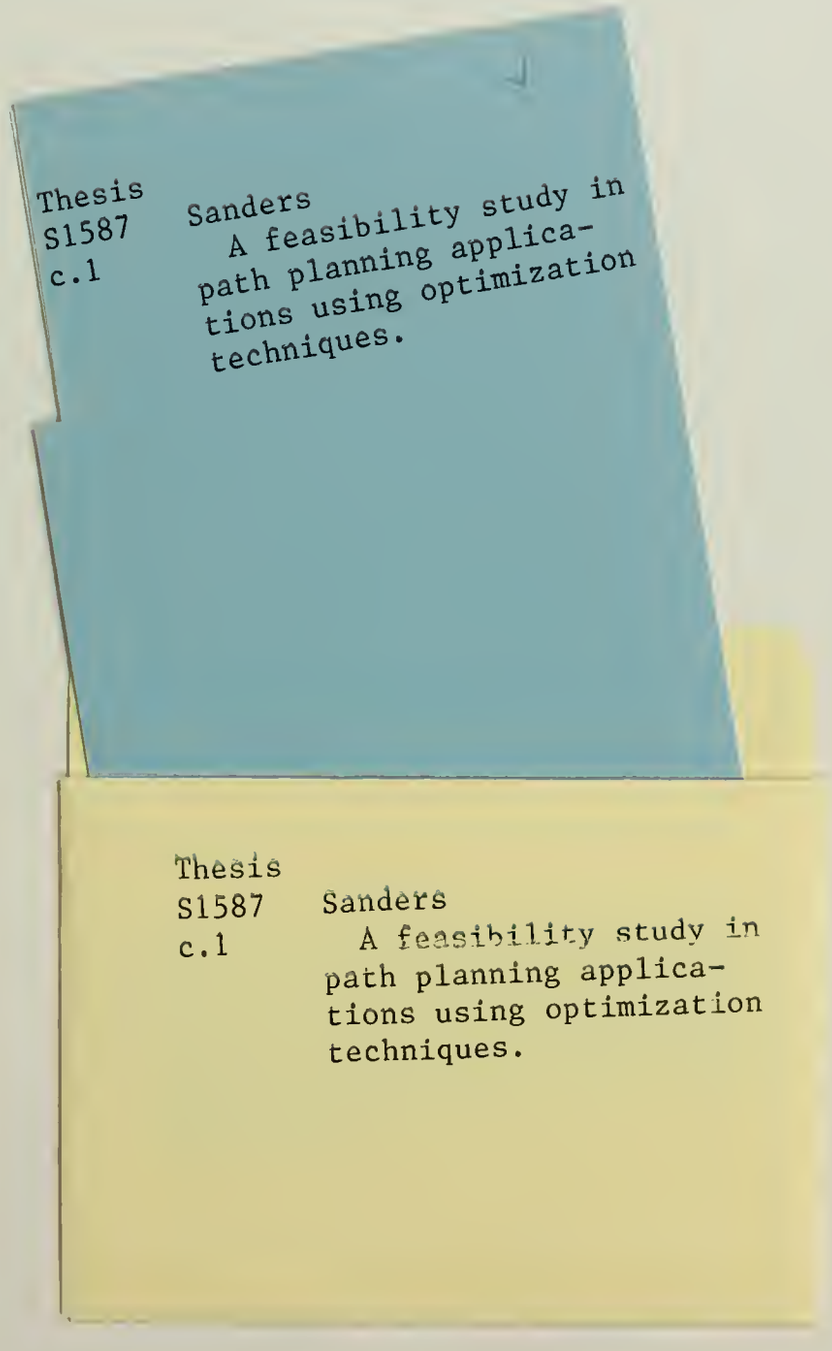
		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Chairman, Code 69Hy Mechanical Engineering Department Naval Postgraduate School Monterey, California 93943-5004	1
4.	Professor D.L. Smith, Code 69Sm Mechanical Engineering Department Naval Postgraduate School Monterey, California 93943-5004	10
5.	Professor R. McGhee, Code 52Mz Computer Science Department Naval Postgraduate School Monterey, California 93943-5004	1
6.	Professor R. Christi, Code 62Cx Electrical and Computer Engineering Department Naval Postgraduate School Monterey, California 93943-5004	1
7.	Dr. G.N. Vanderplaats Engineering Design Optimization, Inc. 1275 Camino Rio Verde Santa Barbara, California 93111	1
8.	LCDR David W. Sanders, USN 204 First Street Butler, Pennsylvania 16001	3
9.	LCDR T. Olson, USN 6905 Ashbury Dr. Springfield, Virginia 22152	1
10.	Russ Weneth, Code u25 Naval Surface Weapons Center White Oak, Maryland 20910	1

11. Paul Heckman, 1
Head, Underseas AI & Robotics Branch
Naval Ocean System Center
San Diego, California 92152

12. RADM G. Curtis, USN PMS-350 1
Naval Sea Systems Command
Washington, D.C. 20362-5101

13. LT Relle L. Lyman, Jr., USN Code 90G 1
Naval Sea Systems Command
Washington, D.C. 20362-5101

14. LT Richard Boncal, USN 1
Raymes Neck Road, RFD 2
York, Maine 03909



AMERICAN BINDERY
TOPEKA KANSAS

thesS1587
A feasibility study in path planning app
3 2768 000 78630 5
DUDLEY KNOX LIBRARY c.1